MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

DANH MỤC THUẬT NGỮ TIẾNG ANH

STT	Thuật ngữ	Diễn giải
1	Tester	Kiểm thử viên, chuyên viên kiểm thử
2	QC	Kiểm soát chất lượng
3	QA	Đảm bảo chất lượng
4	ISO	Tổ chức Tiêu chuẩn hóa Quốc tế
5	IEC	Hội đồng Điện, Điện tử Quốc tế
6	Apache	Phần mềm web server miễn phí mã nguồn mở
7	MySQL	Hệ quản trị cơ sở dữ liệu
8	Grant	Gán quyền
9	Connect	Kết nối
10	Setup	Cài đặt phần mềm
11	Project	Dự án phần mềm
12	Requirement	Yêu cầu người dùng
13	Test Suite	Bộ kiểm thử
14	Test Case	Trường hợp kiểm thử, ca kiểm thử
15	Release	Phát hành phiên bản phần mềm
16	Test Plan	Kế hoạch kiểm thử
17	Report	Bảng báo cáo
18	Metric	Độ đo

CHƯƠNG 1: TỔNG QUAN VỀ KIỂM SOÁT CHẤT LƯỢNG PHẦN MỀM

1. ĐẶT VẤN ĐỀ

1.1. Các hoạt động chính trong đảm bảo chất lượng phần mềm

Các hoạt động chính bao gồm: SQA, SQC và Testing. Mỗi loại hoạt động đều có vai trò khác nhau trong thực hiện quản lý chất lượng phần mềm. Các hoạt động này có những vai trò cụ thể như sau:

- SQA (Software Quality Assurance): là tập hợp các hoạt động được lập ra để đảm bảo tiến trình phát triển và/hoặc duy trì là phù hợp để chắc chắn một hệ thống sẽ đáp ứng các mục tiêu của nó. Chịu trách nhiệm toàn bộ về tiêu chuẩn, quy trình kiểm tra để đảm bảo chất lượng. Có nhiệm vụ giám sát các tiêu chuẩn và quy trình sản xuất phần mềm được định nghĩa và tuân thủ nghiêm túc.

- SQC (Software Quality Control): là tập hợp các hoạt động được tạo ra để đánh giá sản phẩm đang được tiến hành. Có quyền kiểm tra sản phẩm theo phương pháp, tiêu chuẩn nào? Sẽ dùng dụng cụ gì để kiểm tra, và sản phẩm phải đạt được mức độ nào thì sẽ được công nhận là chính phẩm. Khuyết tật nào sẽ quy ra là thứ phẩm,… Trực tiếp kiểm tra chất lượng của sản phẩm. Có nhiệm vụ khảo sát, chạy thử và báo cáo lỗi.

- Testing: là quá trình thực thi hệ thống với ý định tìm kiếm các thiếu sót của phần mềm. Tại một số công ty, Tester đồng nghĩa với SQC. Tuy nhiên, Testing tuy hao hao giống SQC nhưng vẫn có sự khác biệt. SQC nhìn chung có khuynh hướng "confirm" (nghĩa là sản phẩm có làm đúng theo yêu cầu hay không, chấm hết, trong khi Testing nhìn chung có khuynh hướng khám phá để "break" , để tìm lỗi (nghĩa là tìm xem sản phẩm chạy sai như thế nào). Cơ bản, Testing là một hoạt động thuộc SQC do đó nhiều nơi kết hợp SQC và Testing thành một và gọi chung là QC hay Testing team.

Bảng 1: Bảng so sánh các hoạt động trong lĩnh vực đảm bảo chất lượng phần mềm

SQA (Đảm bảo chất lượng)	SQC (Kiểm soát chất lượng)	Testing (Kiểm thử)
SQA bao gồm các hoạt động nhằm đảm bảo các quy trình và tiêu chuẩn được tuân thủ nhằm đảm bảo phần mềm được phát triển theo dự định	SQC bao gồm các hoạt động nhằm xác minh một phần mềm được phát triển tuân thủ các yêu cầu đã được liệt kê	Testing bao gồm các hoạt động nhằm tìm lỗi hoặc khiếm khuyết trong sản phẩm
SQA thì tập trung vào qui trình	SQC thì tập trung vào sản phẩm	Testing thì tập trung vào sản phẩm

1.2. Nghề nghiệp trong quản lý chất lượng phần mềm

1.2.1. Các loại nghề nghiệp phổ biến

Các hoạt động trình bày trên đã tạo ra 03 loại nghề nghiệp, cụ thể:

- Nhân viên QA: là người chịu trách nhiệm đảm bảo chất lượng sản phẩm thông qua việc đưa ra quy trình làm việc giữa các bên liên quan. Nhiệm vụ chính của nhân viên QA:

+ Đề xuất, đưa ra quy trình phát triển sản phẩm phù hợp với các yêu cầu cụ thể của từng dự án.

+ Đưa ra tài liệu, biểu mẫu, hướng dẫn để đảm bảo chất lượng cho tất cả các bộ phận trong nhóm phát triển sản phẩm.

+ Nhắc nhở đội ngũ phát triển sản phẩm trong việc tuân thủ quy trình đã đưa ra

+ Điều chỉnh, thay đổi quy trình phù hợp với từng sản phẩm mà các team đang thực hiện.

- Nhân viên QC: là người chịu trách nhiệm thực hiện công việc kiểm tra chất lượng phần mềm. Nhiệm vụ chính của nhân viên QC:

+ Tìm hiểu hệ thống, phân tích tài liệu mô tả về hệ thống, thiết kế các testcase và thực hiện việc test phần mềm trước khi giao cho khách hàng.

+ Lên kế hoạch kiểm thử.

+ Sử dụng các công cụ kiểm thử để tạo và thực hiện các Test Case.

+ Phối hợp nhóm lập trình trong việc sửa lỗi.

- Nhân viên kiểm thử (Tester): là người lục tung hết các ngóc ngách để tìm càng nhiều lỗi càng tốt, hay còn gọi là test kiểu khám phá phần mềm. Tester thường đóng vai trò người dùng cuối để dùng sản phẩm và tìm lỗi trên sản phẩm, đánh giá những rủi ro tiềm ẩn có thể ảnh hưởng đến chất lượng sản phẩm mà các bên liên quan có thể không lường trước được hay không được nêu ra trong yêu cầu sản phẩm.

1.2.2. So sánh giữa nhân viên QC và Tester

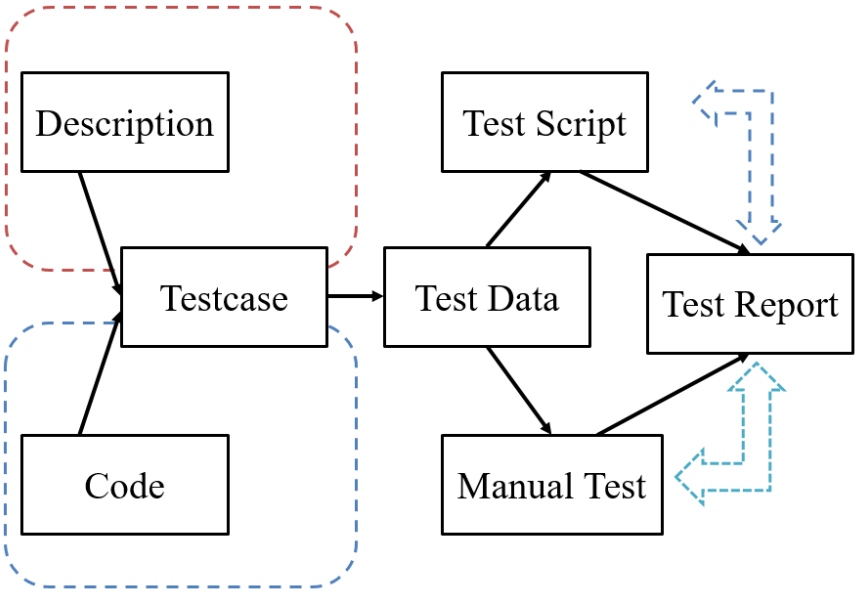
Tester là nhân viên hiểu biết về công nghệ thông tin chuyên phụ trách một hoặc nhiều hoạt động kiểm tra trong sản phẩm phần mềm. Các hoạt động của Tester:

- Xây dựng Test Case.

- Tạo các giá trị cho các Test Case.

- Chạy tập lệnh kiểm thử.

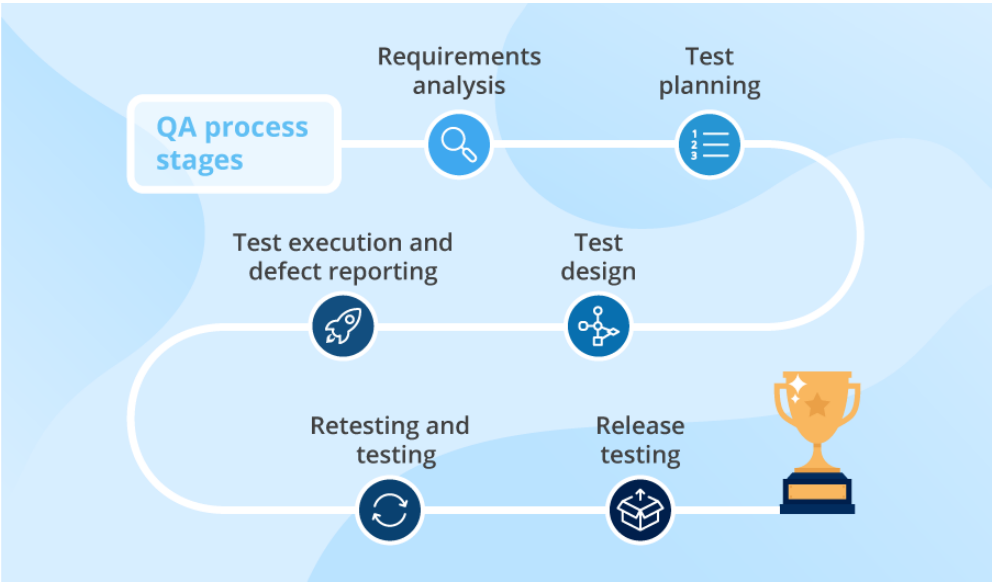
- Phân tích và báo cáo kết quả kiểm thử.



Hình 1: Các hoạt động cơ bản của Tester

Nhân viên QC là nhân viên hiểu biết về công nghệ thông tin chuyên phụ trách một hoặc nhiều hoạt động kiểm tra trong sản phẩm phần mềm. Trực tiếp kiểm tra sản phẩm trước khi bàn giao khách hàng. Các hoạt động của nhân viên QC:

- Tìm hiểu hệ thống (phân tích tài liệu mô tả hệ thống).
- Lên kế hoạch kiểm thử (thiết kế Test Plan).
- Xây dựng Test Case.
- Tạo các giá trị cho các Test Case.
- Chạy tập lệnh kiểm thử.
- Phân tích và báo cáo kết quả kiểm thử.



Hình 2: Các hoạt động cơ bản của nhân viên QC

2. CHẤT LƯỢNG PHẦN MỀM

2.1. Tổng quan về chất lượng

Chất lượng: là khả năng đáp ứng toàn diện nhu cầu của người dùng về tính năng cũng như công dụng được nêu ra một cách tường minh hoặc không tường minh trong những ngữ cảnh xác định (Theo ISO-8402).

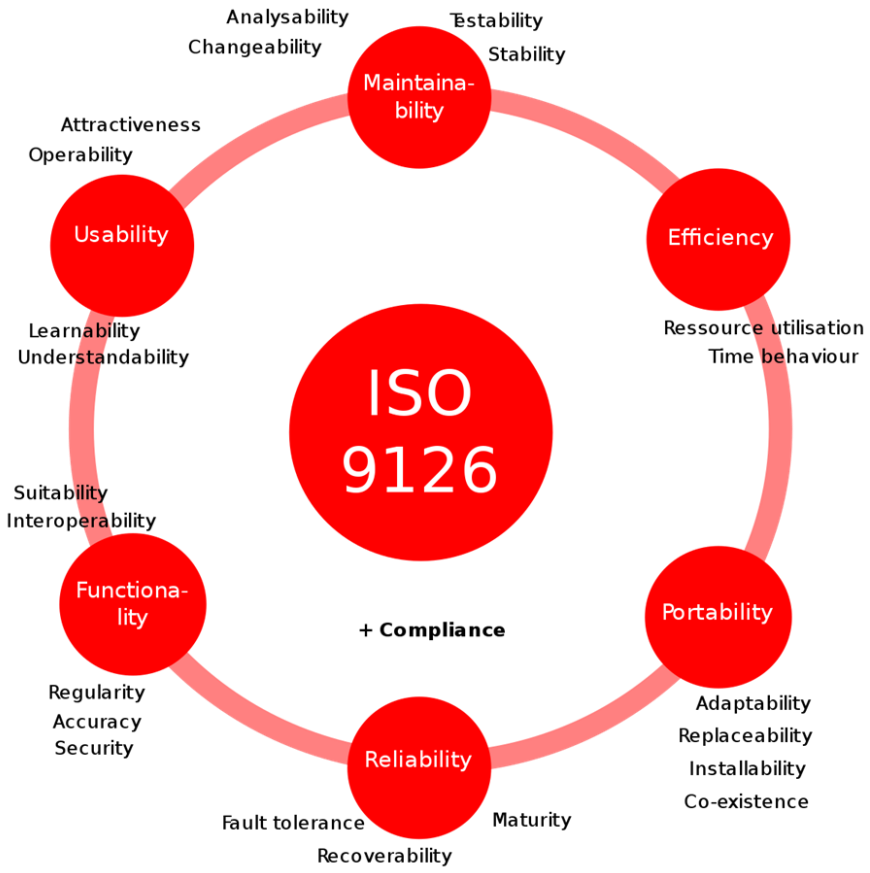
Kiểm soát chất lượng phần mềm: là một phần của đảm bảo chất lượng nhưng sẽ tập trung chủ yếu vào các yêu cầu về chất lượng. Hiểu một cách đơn giản đó là việc người làm sẽ thực hiện quá trình kiểm soát quy trình tạo ra sản phẩm dịch tự bằng nhiều yếu tố. Mục đích để có thể đảm bảo cho ra đời những sản phẩm chất lượng tốt nhất.

Theo cách tiếp cận của ISO, chất lượng toàn diện của phần mềm cần phải được quan tâm từ chất lượng quy trình, tới chất lượng phần mềm nội bộ (chất lượng trong), chất lượng phần mềm đối chiếu với yêu cầu của người dùng (chất lượng ngoài) và chất lượng phần mềm trong sử dụng (chất lượng sử dụng). Mô hình ISO 9126 quy định các loại chất lượng này.

2.2. Mô hình chất lượng

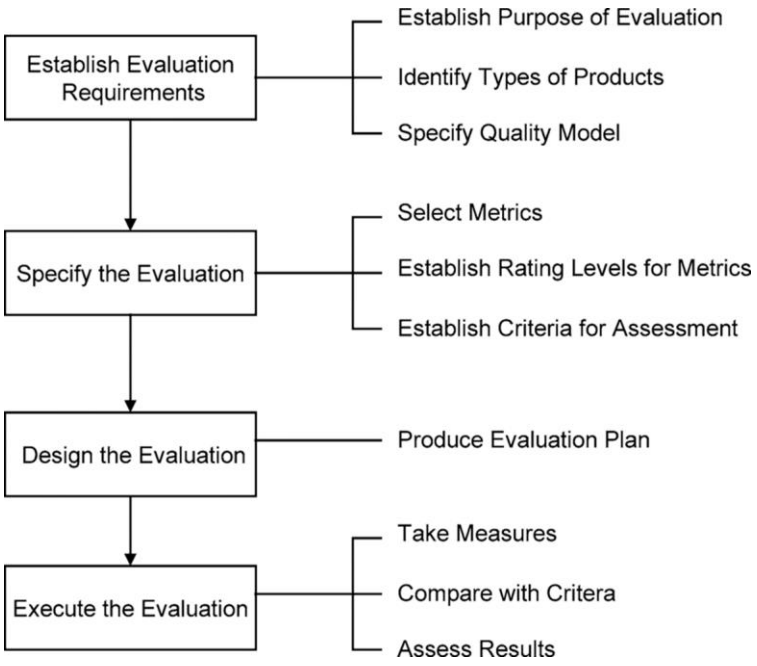
Trong bài giảng này, nhóm tác giả đề cập đến 02 mô hình bao gồm mô hình ISO 9126 và mô hình ISO 14598, cụ thể:

- Mô hình ISO 9126 là tiêu chuẩn quốc tế để đánh giá phần mềm. Nó đưa ra các tiêu chí về chất lượng phần mềm (Chất lượng trong, chất lượng ngoài và Chất lượng sử dụng).



Hình 3: Các tiêu chí trong mô hình 9126

- Mô hình ISO/IEC 14598 đưa ra quy trình đánh giá chất lượng phần mềm. Bên cạnh đó mô hình ISO/IEC 9126 và ISO/IEC 14598 thiết lập một bộ tiêu chuẩn bổ trợ lẫn nhau để kiểm soát chất lượng phần mềm. Mô hình ISO/IEC 9126 được đưa vào tiêu chí Select Metrics trong mô hình ISO/IEC 14598.



Hình 4: Mô hình ISO/IEC 14598

CHƯƠNG 2: TIÊU CHUẨN ISO/IEC 9126

1. GIỚI THIỆU MÔ HÌNH

ISO/IEC 9126 (gọi tắt là ISO 9126) thiết lập một mô hình chất lượng chuẩn cho các sản phẩm phần mềm. Bộ tiêu chuẩn này được chia làm bốn phần:

Một là, ISO 9126-1 đưa ra mô hình chất lượng sản phẩm phần mềm.

Hai là, ISO 9126-2 quy định phép đánh giá chất lượng ngoài.

Ba là, ISO 9126-3 quy định phép đánh giá chất lượng trong.

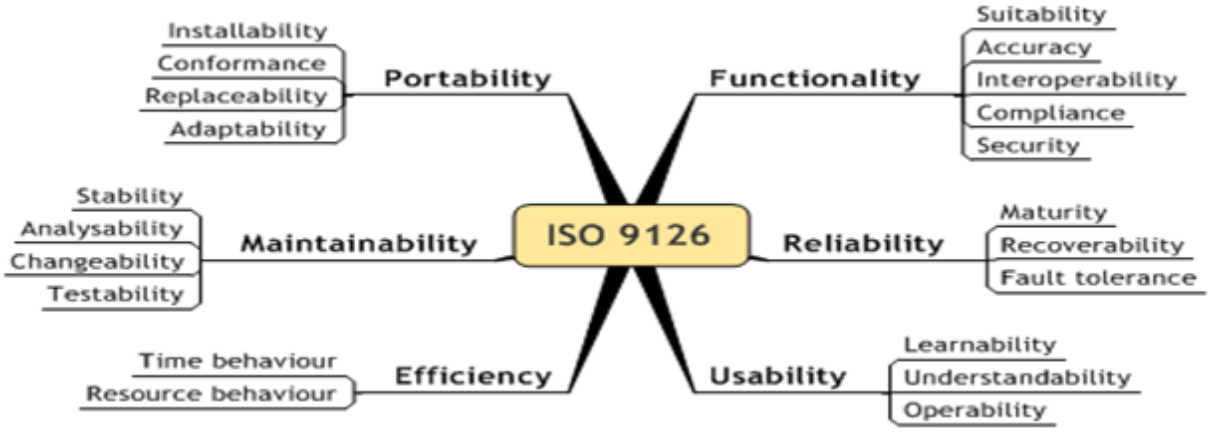
Bốn là, ISO 9126-4 quy định phép đánh giá chất lượng sản phẩm phần mềm trong quá trình sử dụng.

2. PHẠM VI MÔ HÌNH

Phần 1 của mô hình xác định 06 tiêu chí của chất lượng trong và chất lượng ngoài; các tiêu chí này sau đó lại được chia nhỏ thành nhiều tiêu chí con. Phần 2 của mô hình mô tả 4 tiêu chí chất lượng sử dụng.

Mô hình ISO 9126 có thể được dùng để:

- Kiểm tra tính đáp ứng đối với những yêu cầu đã đặt ra.
- Xác định các yêu cầu phần mềm.
- Xác định các đối tượng thiết kế phần mềm.
- Xác định các đối tượng kiểm thử phần mềm.
- Xác định các tiêu chuẩn đảm bảo chất lượng.
- Xác định các tiêu chuẩn chấp nhận cho một sản phẩm phần mềm hoàn chỉnh



Hình 5: Các tiêu chí trong mô hình ISO 9126

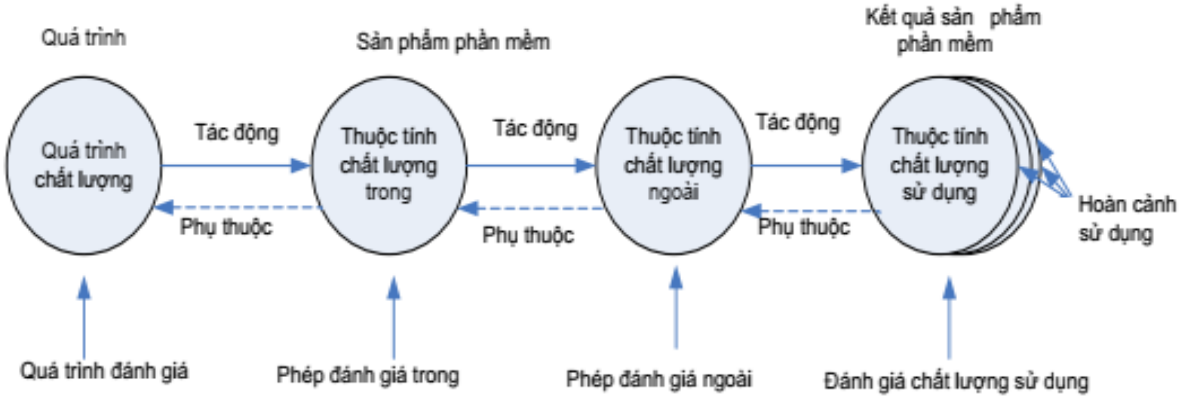
3. TIÊU CHÍ CHẤT LƯỢNG

Chất lượng trong: là tổng hợp của tất cả các đặc điểm của sản phẩm phần mềm từ góc độ của người phát triển phần mềm. Chất lượng trong được đo lường và đánh giá theo các yêu cầu chất lượng trong (sử dụng các phép đánh giá trong).

Chất lượng ngoài: là toàn bộ các đặc điểm của sản phẩm phần mềm từ góc độ của người đánh giá phần mềm độc lập. Chất lượng này thể hiện khi phần mềm hoạt động, nó được đánh giá trong môi trường với dữ liệu giả lập (sử dụng công cụ đánh giá độc lập).

Chất lượng sử dụng: là cách nhìn của người dùng về chất lượng của sản phẩm phần mềm khi nó được sử dụng trong một môi trường và hoàn cảnh cụ thể. Người sử dụng chỉ đánh giá các tiêu chí của phần mềm mà họ dùng tới.

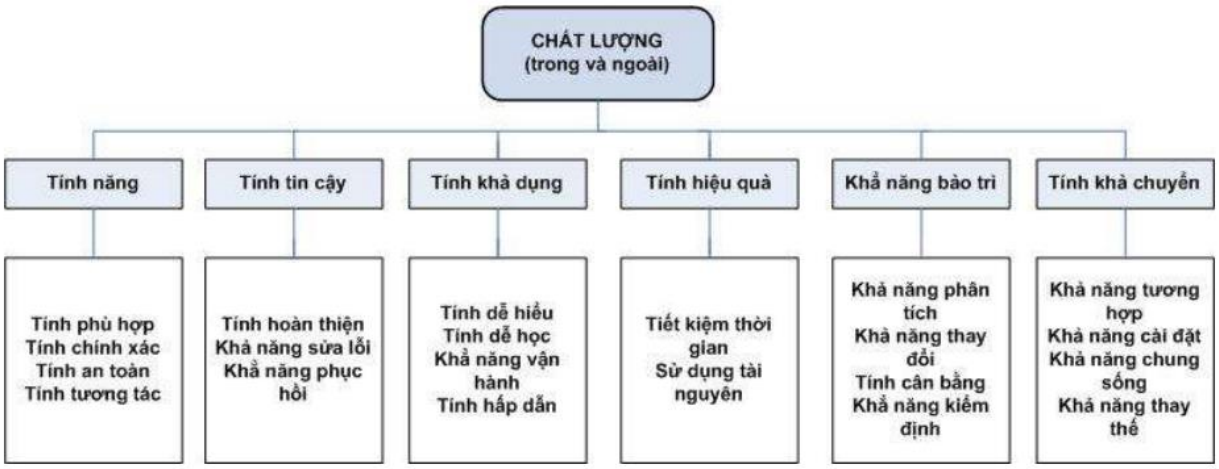
Mối quan hệ giữa các phép đánh giá được thể hiện qua Hình 6.



Hình 6: Mối quan hệ giữa các phép đánh giá

4. MÔ HÌNH CHẤT LƯỢNG TRONG VÀ CHẤT LƯỢNG NGOÀI

Trong mô hình ISO 9126, mô hình chất lượng trong và chất lượng ngoài có các tiêu chí đánh giá giống nhau (Hình 7).



Hình 7: Chất lượng trong và ngoài của mô hình ISO 9126

Chi tiết tiêu chí chất lượng trong và ngoài được nhóm tác giả trình bài trong nội dung bên dưới.

4.1. Tính năng (tính chức năng)

Khả năng của phần mềm cung cấp các chức năng đáp ứng được nhu cầu sử dụng khi phần mềm làm việc trong điều kiện cụ thể.

4.1.1. Tính phù hợp

Phép đánh giá tính phù hợp phải có khả năng đo thuộc tính như sự xuất hiện của chức năng không thỏa mãn hoặc sự xuất hiện của việc vận hành không thỏa mãn trong quá trình kiểm tra và vận hành của người sử dụng của hệ thống.

Chức năng hoặc vận hành không thỏa mãn có thể là :

- Các chức năng và vận hành không hoạt động như trong hướng dẫn sử dụng hoặc đặc tả yêu cầu.

- Các chức năng và vận hành không đưa ra kết quả hợp lý và chấp nhận được để đạt được mục tiêu cụ thể đã định của nhiệm vụ người sử dụng.

Bảng 2: Bảng các phép đánh giá tính phù hợp

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Tính đầy đủ chức năng	Các chức năng được đánh giá được đáp ứng đầy đủ như thế nào?	So sánh số lượng các chức năng thực hiện các nhiệm vụ xác định và số lượng các chức năng được đánh giá	X = 1 - A/B A= Số lượng các chức năng có lỗi phát hiện khi đánh giá B= Số lượng các chức năng được đánh giá	Càng gần 1.0 thì càng được thỏa mãn	Người phát triển SQA
Tính hoàn thiện của triển khai chức năng	Việc triển khai chức năng đối với các đặc tả yêu cầu được hoàn thiện như thế nào?	Thực hiện các kiểm tra chức năng (Kiểm tra kiểu hộp đen) của hệ thống theo các đặc tả yêu cầu. Đếm số lượng các chức năng bị thiếu được phát hiện trong quá trình đánh giá và so sánh với số lượng các chức năng được mô tả trong đặc tả yêu cầu.	X = 1 - A/B A = Số lượng các chức năng bị thiếu được phát hiện trong quá trình đánh giá B = Số lượng các chức năng được mô tả trong đặc tả yêu cầu	Càng gần 1.0 thì càng được thỏa mãn	Người phát triển SQA

Mức độ bao phủ của triển khai chức năng	Việc triển khai chức năng chính xác đến mức nào?	Thực hiện các bài kiểm tra chức năng (kiểm tra kiểu hộp đen) của hệ thống theo các đặc tả yêu cầu. Tính toán số lượng các chức năng thực hiện không đúng hoặc bị thiếu được phát hiện trong quá trình đánh giá và so sánh với số lượng các chức năng được mô tả trong đặc tả yêu cầu.	X=1- A/B A= Số lượng các chức năng triển khai không đúng hoặc chức năng bị thiếu được phát hiện trong quá trình đánh giá. B= Số lượng các chức năng được mô tả trong đặc tả yêu cầu	Càng gần 1.0 thì càng được thỏa mãn	Người phát triển SQA
Tính ổn định đặc tính chức năng (tính không ổn định)	Các đặc tính chức năng ổn định như thế nào sau khi đi vào hoạt động	Đếm số lượng các chức năng được mô tả trong đặc tính chức năng bị thay đổi sau khi hệ thống đi vào hoạt động và so sánh với tổng số lượng các chức năng được mô tả trong đặc tả yêu cầu	X=1- A/B A= Số lượng các chức năng bị thay đổi khi đi vào hoạt động bắt đầu từ khi hoạt động B = Số lượng chức năng được mô tả trong đặc tả yêu cầu	Càng gần 1.0 thì càng được thỏa mãn	Người phát triển SQA

4.1.2. Tính chính xác

Phép đánh giá tính chính xác có khả năng đo thuộc tính như tần suất của người sử dụng gặp phải sự xuất hiện của các sự kiện không chính xác.

Các sự kiện không chính xác bao gồm:

- Kết quả không đúng hoặc không chính xác gây ra bởi dữ liệu không thỏa đáng; ví dụ, dữ liệu với quá ít chữ số có nghĩa cho tính toán đúng.

- Trái ngược giữa các thủ tục vận hành thực tế và các mô tả trong hướng dẫn vận hành.

- Khác nhau giữa các kết quả thực tế và kết quả mong đợi hợp lý của các nhiệm vụ thực hiện trong quá trình vận hành.

Bảng 3: Bảng các phép đánh giá tính chính xác

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Độ chính xác mong đợi	Sự khác nhau giữa các kết quả thực và các kết quả mong đợi hợp lý có chấp nhận được không?	Thực hiện các trường hợp kiểm tra chuẩn đầu vào và đầu ra với các kết quả mong đợi hợp lý. Đếm số trường hợp bắt gặp bởi người sử dụng với sự khác biệt không thể chấp nhận được với các kết quả mong đợi hợp lý	X = A/T A = Số trường hợp bắt gặp bởi người sử dụng mà khác biệt với kết quả mong đợi lớn hơn giới hạn cho phép T = Thời gian thực hiện	Càng gần bằng 0 càng tốt	Người phát triển Người sử dụng

Độ chính xác tính toán	Người sử dụng cuối thường gặp phải các kết quả không chính xác như thế nào?	Ghi nhận số lượng các tính toán không chính xác dựa trên các đặc tính	X = A/T A = Số các tính toán không chính xác người sử dụng gặp phải T = Thời gian thực hiện	Càng gần bằng 0 càng tốt	Người phát triển Người sử dụng
Độ chính xác	Người sử dụng cuối thường gặp phải các kết quả với độ chính xác không thỏa mãn như thế nào?	Ghi nhận số lượng các kết quả với độ chính xác không thỏa mãn	X = A/T A = Số các kết quả người sử dụng gặp phải với độ chính xác khác với yêu cầu T = Thời gian thực hiện	Càng gần bằng 0 càng tốt	Người phát triển Người sử dụng

4.1.3. Tính an toàn

Phép đánh giá tính an toàn có khả năng đo thuộc tính như số chức năng cùng với, hoặc sự xuất hiện các vấn đề an toàn, chúng là:

- Thất bại ngăn chặn lỗ hổng của thông tin hay dữ liệu cần bảo vệ;
- Thất bại ngăn chặn mất dữ liệu quan trọng;
- Thất bại bảo vệ chống lại truy cập bất hợp pháp hay vận hành bất hợp pháp.

Bảng 4: Bảng các phép đánh giá tính an toàn

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Khả năng kiểm toán truy cập	Theo dõi kiểm toán truy cập liên quan đến truy cập của người sử dụng vào hệ thống và dữ liệu được hoàn thành như thế nào?	Đánh giá lượng truy cập được hệ thống ghi lại trong cơ sở dữ liệu lược sử truy cập.	X = A/B A = Số lượng "người sử dụng truy cập vào hệ thống và dữ liệu" được ghi trong cơ sở dữ liệu lược sử truy cập B = Số lượng "người sử dụng truy cập vào hệ thống và dữ liệu" được thực hiện trong quá trình đánh giá	Càng gần bằng 1 càng tốt	Người phát triển

Ngăn chặn sai lạc dữ liệu	Tuần sát của các sự kiện sai lạc dữ liệu là bao nhiêu?	Đếm sự xuất hiện của các sự kiện sai lạc dữ liệu lớn và nhỏ	a) X = 1 -A/N A = Số lần sự kiện sai lạc dữ liệu lớn xảy ra N = Số trường hợp kiểm tra chuẩn cố gắng gây ra sai lạc dữ liệu b) Y = 1 - B/N B = Số lần sự kiện sai lạc dữ liệu nhỏ xảy ra c) Z = A/T hoặc B/T T = Khoảng thời gian thực hiện (trong quá trình kiểm tra)	Càng gần bằng 1 càng tốt Càng gần bằng 1 càng tốt Càng gần 0 càng tốt	Người bảo trì Người phát triển

4.1.4. Tính tương tác

Phép đánh giá khả năng tương tác có khả năng đo thuộc tính như số lượng các chức năng hoặc sự xuất hiện tính kém liên hệ bao gồm dữ liệu và lệnh, mà chúng được truyền tải trước đó giữa sản phẩm phần mềm và các hệ thống khác, các sản phẩm phần mềm khác, hoặc thiết bị được kết nối.

Bảng 5: Bảng các phép đánh giá khả năng tương tác

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Khả năng trao đổi dữ liệu (Dựa trên định dạng dữ liệu)	Các chức năng giao diện trao đổi cho dữ liệu xác định chuyển giữa các hoạt động được thực hiện chính xác như thế nào?	Kiểm tra mỗi định dạng bản ghi đầu ra của chức năng giao diện đường xuống của hệ thống theo các đặc tính trường dữ liệu. Đếm số định dạng dữ liệu được chấp thuận để trao đổi với phần mềm hoặc hệ thống khác trong quá trình kiểm tra trao đổi dữ liệu so với tổng số.	X = A/B A = Số lượng định dạng dữ liệu được chấp thuận trao đổi thành công với phần mềm hệ thống khác trong quá trình kiểm tra trao đổi dữ liệu. B = Tổng số định dạng dữ liệu được trao đổi	Càng gần bằng 1 càng tốt	Người phát triển

Khả năng trao đổi dữ liệu (Dựa vào cố gắng thực hiện thành công của người sử dụng)	Người sử dụng thường trao đổi dữ liệu thất bại giữa phần mềm đích và phần mềm khác như thế nào? Truyền dữ liệu giữa phần mềm đích và phần mềm khác thường thành công như thế nào? Liệu người sử dụng có thường thành công trong trao đổi dữ liệu không	Đếm số trường hợp chức năng giao diện được sử dụng và bị lỗi.	a) X = 1 - A/B A = Số trường hợp người sử dụng thất bại khi truyền dữ liệu với phần mềm hoặc hệ thống khác B = Số trường hợp người sử dụng cố gắng trao đổi dữ liệu b) Y = A/T T = Khoảng thời gian thực hiện	Càng gần bằng 1 càng tốt Càng gần 0 càng tốt	Người bảo trì

4.2. Tính tin cậy (Độ ổn định)

Phép đánh giá tính tin cậy phải có khả năng đo các thuộc tính liên quan tới hoạt động của hệ thống mà phần mềm là một phần của nó trong quá trình thực hiện kiểm tra để chỉ ra khả năng của tính tin cậy của phần mềm trong hệ thống trong suốt quá trình vận hành. Các hệ thống và phần mềm không phân biệt với nhau trong phần lớn trường hợp.

4.2.1. Tính hoàn thiện

Phép đánh giá tính hoàn thiện có khả năng đo các thuộc tính như phần mềm không gặp sự cố gây ra bởi lỗi tồn tại trong chính bản thân phần mềm .

Bảng 6: Bảng các phép đánh giá tính hoàn thiện

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Ước lượng mật độ lỗi tiềm tàng	Có bao nhiêu vấn đề vẫn tồn tại có thể xảy ra như các lỗi tương lai?	Đếm số lượng các lỗi được phát hiện trong giai đoạn thử nghiệm xác định và dự báo số lượng tiềm tàng các lỗi tương lai sử dụng mô hình ước lượng tăng tính tin cậy	X = {ABS(A1-A2)}/B (X = Mật độ lỗi tiềm tàng còn lại được ước lượng) ABS()= Giá trị tuyệt đối A1= Số lượng tổng các lỗi tiềm tàng được dự báo trong sản phẩm phần mềm A2= Tổng số lỗi phát hiện được trong thực tế B= Kích cỡ sản phẩm	Phụ thuộc vào giai đoạn kiểm tra. Càng về giai đoạn cuối, các giá trị càng nhỏ càng tốt	Người phát triển Người kiểm tra SQA Người sử dụng

Mật độ lỗi đối chiếu với các trường hợp kiểm tra chuẩn	Có bao nhiêu lỗi được phát hiện trong giai đoạn thử nghiệm xác định?	Đếm số lượng các lỗi được phát hiện và số các trường hợp kiểm tra chuẩn được thực hiện	X = A1/A2 A1 = Số lượng các lỗi được phát hiện A2 = Số lượng các trường hợp kiểm tra chuẩn được thực hiện	Phụ thuộc vào giai đoạn kiểm tra. Càng về giai đoạn cuối, các giá trị càng nhỏ càng tốt	Người phát triển Người kiểm tra SQA
Giải quyết lỗi	Có bao nhiêu trường hợp lỗi được giải quyết?	Đếm số lượng các lỗi không xuất hiện lại trong giai đoạn thử nghiệm xác định dưới các điều kiện tương tự nhau. Duy trì báo cáo giải quyết các vấn đề mô tả tình trạng của tất cả các lỗi.	X = A1/A2 A1 = Số lượng các lỗi được giải quyết A2 = Tổng số các lỗi được phát hiện trong thực tế	Càng tiến tới 1.0 càng tốt, vì càng có nhiều lỗi được giải quyết.	Người sử dụng SQA Người bảo trì
Mật độ lỗi	Có bao nhiêu lỗi được phát hiện trong giai đoạn thử nghiệm xác định?	Đếm số lượng các lỗi được phát hiện và tính toán mật độ.	X = A/B A1 = Số lượng các lỗi được phát hiện B = Kích thước sản phẩm	Càng về giai đoạn cuối, các giá trị càng nhỏ càng tốt.	Người phát triển Người kiểm tra SQA
Loại bỏ lỗi	Có bao nhiêu lỗi đã được chỉnh sửa?	Đếm số lượng các lỗi được loại bỏ trong quá trình kiểm thử và so sánh với tổng số lỗi được phát hiện và tổng số lỗi được dự báo	a) X = A1/A2 A1 = Số lượng các lỗi được chỉnh sửa A2 = Tổng số lỗi được phát hiện thực tế b) Y = A1/A3 A3 = Tổng số lỗi tiềm tàng được dự báo trong sản phẩm phần mềm	Càng tiến tới 1.0 càng tốt vì càng có ít lỗi còn tồn tại.	Người phát triển SQA Người bảo trì

Thời gian trung bình giữa các lỗi (MTBF)	Phần mềm thường hay gặp lỗi trong quá trình hoạt động như thế nào?	Đếm số lượng các lỗi xuất hiện trong một giai đoạn nhất định của quá trình hoạt động và tính toán khoảng thời gian trung bình giữa các lỗi.	a) X = T1/A b) Y = T2/A T1 = Thời gian hoạt động T2 = Tổng thời gian giữa các lần xuất hiện lỗi liên tiếp A = Tổng số lượng lỗi được phát hiện thực tế (Các lỗi xuất hiện trong thời gian hoạt động được quan sát)	Càng lớn càng tốt. Vì rằng thời gian mong đợi giữa các lỗi càng dài.	Người bảo trì Người sử dụng
Tính bao phủ kiểm tra (Tính bao phủ của quá trình kiểm tra trong kịch bản hoạt động xác định)	Bao nhiêu trường hợp kiểm tra chuẩn yêu cầu đã được thực hiện trong quá trình kiểm tra?	Đếm số lượng các trường hợp kiểm tra chuẩn được thực hiện trong quá trình kiểm tra và so sánh với số trường hợp kiểm tra chuẩn yêu cầu để đạt được mức bao phủ kiểm tra thích hợp.	X = A/B A = Số lượng các trường hợp kiểm tra chuẩn được thực hiện thực tế tiêu biểu cho kịch bản hoạt động trong quá trình kiểm tra B = Số lượng các trường hợp kiểm tra chuẩn cần thực hiện để bao phủ các yêu cầu	Càng gần 1.0 tính bao phủ của kiểm tra càng tốt	Người phát triển Người kiểm tra SQA
Tính kỹ lưỡng của kiểm tra	Liệu sản phẩm đã được kiểm tra cẩn thận?	Đếm số lượng các trường hợp kiểm tra chuẩn thành công đã được thực hiện thực tế và so sánh nó với tổng số lượng các trường hợp kiểm tra chuẩn được hoàn tất cho từng loại yêu cầu.	X = A/B A = Số lượng các trường hợp kiểm tra chuẩn thành công trong quá trình kiểm tra hay hoạt động. B = Số lượng các trường hợp kiểm tra chuẩn được hoàn tất để bao phủ các yêu cầu.	Càng tiến tới 1.0 càng tốt	Người phát triển Người kiểm tra SQA

4.2.2. Khả năng sửa lỗi (khả năng chịu lỗi)

Phép đánh giá khả năng chịu lỗi liên quan đến khả năng phần mềm bảo trì mức hiệu năng nhất định trong các trường hợp lỗi vận hành hoặc vi phạm giao diện xác định của nó.

Bảng 7: Bảng các phép đánh giá khả năng chịu lỗi

Tên phép đánh giá	Mục đích của phép đánh giá	Phương pháp áp dụng	Phép đo, công thức và tính toán các thành phần dữ liệu	Chuyển đổi giá trị đo	Đối tượng sử dụng
Tránh hỏng hóc	Sản phẩm phần mềm thường xuyên gây ra hỏng hóc cho toàn bộ môi trường sản xuất như thế nào?	Đếm số hỏng hóc xảy ra đối với số lượng lỗi. Nếu lỗi xảy ra trong quá trình vận hành thì phân tích bản ghi lược sử hoạt động của người sử dụng	X = 1-A/B A = Số hỏng hóc B = Số lỗi	Càng gần bằng 1.0 càng tốt	Người sử dụng Người bảo trì
Tránh lỗi	Có bao nhiêu mẫu lỗi được đặt dưới sự kiểm soát để tránh các lỗi then chốt và nghiêm trọng?	Đếm số lượng mẫu lỗi đã được tránh và so sánh nó với số mẫu lỗi được xem xét.	X = A/B A = Số lần xuất hiện lỗi then chốt và nghiêm trọng tránh được trong các trường hợp kiểm tra chuẩn của các mẫu lỗi B = Số trường hợp kiểm tra chuẩn mẫu lỗi được tiến hành (hầu như gây ra lỗi) trong quá trình kiểm tra.	Càng gần bằng 1.0 càng tốt, vì người sử dụng có thể tránh được các lỗi then chốt và nghiêm trọng nhiều hơn	Người sử dụng Người bảo trì
Tránh vận hành sai	Có bao nhiêu chức năng được thực thi với khả năng tránh vận hành không đúng?	Đếm số các trường hợp kiểm tra chuẩn của vận hành sai đã được tránh gây ra những lỗi then chốt và nghiêm trọng và so sánh nó với số các trường hợp kiểm tra chuẩn đã được thực hiện của các mẫu vận hành sai được xem xét.	X = A/B A = Số lỗi then chốt và và nghiêm trọng đã được tránh B = Số trường hợp kiểm tra chuẩn được thực hiện của các mẫu vận hành sai (gần như chắc chắn sẽ tạo lỗi) trong quá trình kiểm tra	Càng gần bằng 1.0 càng tốt, vì càng nhiều vận hành sai của người sử dụng được tránh	Người sử dụng Người bảo trì

4.2.3. Khả năng phục hồi

Phép đánh giá khả năng phục hồi có khả năng đo các thuộc tính như phần mềm cùng hệ thống có khả năng thiết lập lại mức hiệu năng thỏa đáng và phục hồi dữ liệu trực tiếp ảnh hưởng trong trường hợp sự cố.

Bảng 8: Bảng các phép đánh giá khả năng phục hồi

Tên phép đánh giá	Mục đích của phép đánh giá	Phương pháp áp dụng	Phép đo, công thức và tính toán các thành phần dữ liệu	Chuyển đổi giá trị đo	Đối tượng sử dụng
Khả năng sẵn sàng	Mức độ sẵn sàng sử dụng của hệ thống trong khoảng thời gian xác định như thế nào?	Hệ thống kiểm tra trong môi trường giống như quá trình sản xuất trong khoảng thời gian xác định thực hiện tất cả các thao tác của người sử dụng. Đo khoảng thời gian sửa chữa mỗi khi hệ thống không sẵn sàng trong quá trình thử nghiệm. Tính toán thời gian trung bình để sửa chữa	a) X = {(To/(To+Tr)} b) Y = A1/A2 To = Thời gian vận hành Tr = Thời gian sửa chữa A1 = Tổng số trường hợp sẵn sàng sử dụng của phần mềm khi sử dụng A2 = Tổng số trường hợp người sử dụng thử sử dụng phần mềm trong thời gian quan sát. Xuất phát từ quan điểm vận hành chức năng người sử dụng	Càng lớn và càng gần 1.0 càng tốt, vì người sử dụng có thể sử dụng phần mềm lâu hơn.	Người sử dụng Người bảo trì
Thời gian chết trung bình	Thời gian trung bình hệ thống không sẵn sàng khi lỗi xảy ra trước khi khởi động lại từ từ là bao nhiêu?	Đo thời gian chết mỗi lần hệ thống không sẵn sàng trong khoảng thời gian thử nghiệm xác định và tính thời gian trung bình	X = T/N T = Tổng thời gian chết N = Số lần hỏng hóc quan sát được Trường hợp xấu nhất và phân bố của thời gian chất cũng phải được đo.	Càng nhỏ càng tốt, vì thời gian hệ thống chết càng nhỏ hơn	Người sử dụng Người bảo trì
Thời gian phục hồi trung bình	Thời gian trung bình mà hệ thống hoàn thành việc phục hồi bắt đầu từ giai đoạn đầu của quá trình phục hồi là bao nhiêu?	Đo thời gian phục hồi toàn bộ cho mỗi lần hệ thống có sự cố trong khoảng thời gian thử nghiệm xác định và tính thời gian trung bình.	X = Tổng (T)/B T = Thời gian phục hồi hệ thống phần mềm bị lỗi tại mỗi thời điểm N = Số trường hợp hệ thống phần mềm được đưa vào phục hồi quan sát được	Càng nhỏ càng tốt	Người sử dụng Người bảo trì

Khả năng khởi động lại	Tần suất hệ thống có thể khởi động lại để cung cấp dịch vụ cho người sử dụng trong một khoảng thời gian yêu cầu là như thế nào?	Đếm số lần hệ thống khởi động lại và cung cấp dịch vụ cho người sử dụng trong khoảng thời gian yêu cầu mục tiêu và so sánh nó với tổng số lần khởi động lại, khi hệ thống có sự cố trong khoảng thời gian thử nghiệm xác định.	X = A/B A = Số lần khởi động lại đáp ứng khoảng thời gian yêu cầu trong quá trình kiểm tra hoặc hỗ trợ vận hành cho người sử dụng B = Tổng số lần khởi động lại trong quá trình kiểm tra hoặc hỗ trợ vận hành cho người sử dụng	Càng lớn và càng gần 1.0 càng tốt, vì người sử dụng có thể khởi động lại dễ dàng.	Người sử dụng Người bảo trì
Khả năng khôi phục lại	Khả năng tự khôi phục lại của sản phẩm sau khi có sự kiện bất thường xảy ra hoặc khi có yêu cầu?	Đếm số lần khôi phục thành công và so sánh với số lần khôi phục được kiểm tra được yêu cầu trong đặc tả. Các ví dụ về yêu cầu khôi phục: điểm kiểm tra của cơ sở dữ liệu, điểm kiểm tra của các giao dịch, chức năng làm ngược thao tác cũ, chức năng thực hiện lại thao tác cũ,…	X = A/B A = Số lượng trường hợp khôi phục được thực hiện thành công B = Số trường hợp khôi phục đã được kiểm tra theo yêu cầu	Càng gần bằng 1.0 càng tốt, vì sản phẩm có khả năng phục hồi tốt hơn trong các trường hợp xác định	Người sử dụng Người bảo trì
Tính hiệu quả của việc khôi phục lại	Khả năng khôi phục lại hiệu quả như thế nào?	Đếm số lần khôi phục được kiểm tra đáp ứng thời gian khôi phục mục tiêu và so sánh với số lần khôi phục cần thiết với thời gian mục tiêu xác định.	X = A/B A = Số trường hợp khôi phục thành công đáp ứng thời gian khôi phục mục tiêu B = Số trường hợp được thực hiện	Càng gần bằng 1.0 càng tốt, vì quá trình khôi phục trong sản phẩm hiệu quả hơn	Người sử dụng Người bảo trì

4.3. Tính khả dụng

Các phép đánh giá tính khả dụng đo khả năng mà phần mềm có thể được hiểu, học, vận hành, thân thiện và tuân thủ với các quy tắc khả dụng và hướng dẫn.

Rất nhiều các phép đánh giá tính khả dụng ngoài được kiểm tra bởi người sử dụng khi thử sử dụng chức năng. Các kết quả sẽ bị tác động bởi các khả năng của người sử dụng và đặc tính của hệ thống. Điều này không phủ nhận tính hợp lệ của các phép đo, do phần mềm được đánh giá thực hiện dưới các điều kiện rõ ràng xác định bởi người sử dụng mẫu đại diện cho nhóm người sử dụng được thừa nhận. Đối với các sản phẩm mục đích chung, các đại diện của một dải các nhóm người sử dụng có thể được dùng. Để kết quả tin cậy, cần mẫu của ít nhất tám người sử dụng, mặc dù thông tin có ích có thể đạt được từ các nhóm nhỏ hơn. Người sử dụng phải hoàn thành kiểm tra không dùng bất cứ gợi ý hay trợ giúp bên ngoài nào.

Các phép đánh giá cho tính dễ hiểu, dễ học, và dễ vận hành có hai dạng phương pháp ứng dụng: kiểm tra người sử dụng hay kiểm tra của sản phẩm khi sử dụng.

4.3.1. Tính dễ hiểu

Người sử dụng phải có khả năng lựa chọn sản phẩm phần mềm phù hợp cho việc sử dụng dự kiến của họ. Phép đánh giá tính dễ hiểu có khả năng ước lượng người sử dụng mới có thể hiểu hay không:

- Phần mềm có phù hợp hay không.
- Nó có thể được sử dụng như thế nào cho các nhiệm vụ đặc thù.

Bảng 9: Bảng các phép đánh giá tính dễ hiểu

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Tính đầy đủ của tài liệu mô tả	Tỷ lệ các chức năng (hoặc các dạng chức năng) hiểu được sau khi đọc tài liệu mô tả sản phẩm là bao nhiêu?	Tiến hành kiểm tra người sử dụng và phỏng vấn người sử dụng với các câu hỏi hoặc quan sát thái độ người sử dụng. Đếm số chức năng được hiểu thỏa đáng và so sánh với tổng số lượng chức năng trong sản phẩm.	X = A/B A = Số chức năng (hay các dạng chức năng) hiểu được B = Tổng số các chức năng (hay các dạng chức năng)	Càng tiến gần tới 1.0 càng tốt	Người sử dụng Người bảo trì
Khả năng truy cập thuyết minh	Tỷ lệ của quá trình thuyết minh/ hướng dẫn người sử dụng có thể truy cập là bao nhiêu?	Tiến hành kiểm tra người sử dụng và quan sát thái độ người sử dụng. Đếm số các chức năng được thuyết minh thỏa đáng và so sánh với tổng số các chức năng yêu cầu khả năng thuyết minh	X = A/B A = Số các thuyết minh/ hướng dẫn mà người sử dụng truy cập thành công B = Số các thuyết minh/ hướng dẫn sẵn có	Càng tiến gần tới 1.0 càng tốt	Người sử dụng Người bảo trì
Khả năng truy cập thuyết minh khi sử dụng	Tỷ lệ của thuyết minh/ hướng dẫn người sử dụng có thể truy cập bất cứ khi nào người sử dụng thực sự cần trong quá trình vận hành là bao nhiêu?	Quan sát thái độ người sử dụng đang cố gắng xem các thuyết minh/hướng dẫn. Việc quan sát có thể được thực hiện thông qua phương thức giám sát hành động có ý thức của con người với máy Camera.	X = A/B A = Số các trường hợp người sử dụng xem được thuyết trình khi họ cố gắng thử xem. B = Số các trường hợp người sử dụng cố gắng xem thuyết trình trong giai đoạn quan sát	Càng tiến gần tới 1.0 càng tốt	Người sử dụng Người bảo trì

Hiệu quả của thuyết minh	Tỷ lệ các chức năng người sử dụng có thể vận hành thành công sau khi được thuyết minh hay hướng dẫn là bao nhiêu?	Quan sát thái độ người sử dụng đang cố gắng xem các thuyết minh/hướng dẫn. Việc quan sát có thể được thực hiện thông qua phương thức giám sát hành động có ý thức của con người với máy Camera.	X = A/B A = Số các chức năng hoạt động thành công B = Số lượng các thuyết minh/ hướng dẫn được truy cập	Càng tiến gần tới 1.0 càng tốt	Người sử dụng Người bảo trì
Các chức năng rõ ràng	Tỷ lệ các chức năng (hoặc các dạng chức năng) có thể được nhận biết bởi người sử dụng dựa trên các điều kiện ban đầu bao nhiêu?	Tiến hành kiểm tra người sử dụng và phỏng vấn người sử dụng với các câu hỏi hoặc quan sát thái độ người sử dụng. Đếm số các chức năng rõ ràng đối với người sử dụng và so sánh với tổng số các chức năng.	X = A/B A = Số lượng các chức năng (hoặc các dạng chức năng) được nhận biết bởi người sử dụng B = Tổng số các chức năng thực tế (hoặc các dạng chức năng)	Càng tiến gần tới 1.0 càng tốt	Người sử dụng Người bảo trì
Khả năng dễ hiểu của chức năng	Tỷ lệ các chức năng sản phẩm người sử dụng có khả năng hiểu được chính xác là bao nhiêu?	Tiến hành kiểm tra người sử dụng và phỏng vấn người sử dụng với các câu hỏi. Đếm số các chức năng giao diện người sử dụng mà mục đích dễ hiểu và so sánh với số các chức năng sẵn sàng cho người sử dụng.	X = A/B A = Số lượng các chức năng giao diện mà mục đích của nó được mô tả chính xác bởi người sử dụng B = Tổng số các chức năng sẵn sàng từ giao diện	Càng tiến gần tới 1.0 càng tốt	Người sử dụng Người bảo trì
Đầu vào và đầu ra có khả năng hiểu được	Người sử dụng có thể hiểu được những gì được yêu cầu tại dữ liệu đầu vào và những gì được cung cấp tại dữ liệu đầu ra bởi hệ thống phần mềm hay không?	Tiến hành kiểm tra người sử dụng và phỏng vấn người sử dụng với các câu hỏi hoặc quan sát thái độ người sử dụng. Đếm số các thành phần dữ liệu đầu vào và đầu ra mà người sử dụng có thể hiểu và so sánh với tổng số các thành phần sẵn sàng cho người sử dụng.	X = A/B A = Số lượng các thành phần dữ liệu đầu vào và đầu ra người sử dụng có thể hiểu chính xác. B = Số lượng các thành phần dữ liệu đầu vào và đầu ra sẵn sàng từ giao diện	Càng tiến gần tới 1.0 càng tốt	Người sử dụng Người bảo trì

4.3.2. Tính dễ học

Phép đánh giá khả năng dễ học phải có khả năng ước lượng người sử dụng mất bao nhiêu thời gian để học sử dụng các chức năng đặc thù, và tính hiệu quả của các hệ thống và tài liệu trợ giúp.

Khả năng dễ học liên hệ chặt chẽ với khả năng dễ hiểu, và các phép đo khả năng dễ hiểu có thể là bộ chỉ thị tiềm năng dễ học của phần mềm.

Bảng 10: Bảng các phép đánh giá tính dễ học

Tên phép đánh giá	Mục đích của phép đánh giá	Phương pháp áp dụng	Phép đo, công thức và tính toán các thành phần dữ liệu	Chuyển đổi giá trị đo	Đối tượng sử dụng
Khả năng dễ dàng học chức năng	Người sử dụng mất bao lâu để học sử dụng chức năng?	Tiến hành kiểm tra người sử dụng và quan sát thái độ người sử dụng.	T = Thời gian trung bình để học sử dụng chức năng chính xác	Càng nhỏ càng tốt	Người sử dụng Người bảo trì
Khả năng dễ dàng học cách thực hiện một nhiệm vụ khi sử dụng	Người sử dụng mất bao nhiêu thời gian để học cách thực hiện một nhiệm vụ xác định một cách hiệu quả	Quan sát thái độ người sử dụng từ khi họ bắt đầu học đến khi họ bắt đầu vận hành một cách hiệu quả.	T = Tổng thời gian khai thác của người sử dụng cho đến khi người sử dụng đạt được thực hiện một nhiệm vụ xác định trong thời gian ngắn	Càng nhỏ càng tốt	Người sử dụng Người bảo trì
Tính hiệu quả của các tài liệu hướng dẫn người sử dụng và/ hoặc hệ thống trợ giúp	Tỷ lệ nhiệm vụ được hoàn thành một cách chính xác sau khi sử dụng tài liệu hướng dẫn và/ hoặc hệ thống trợ giúp là bao nhiêu?	Tiến hành kiểm tra người sử dụng và quan sát thái độ của họ. Đếm số nhiệm vụ được hoàn thành đúng sau khi truy cập vào hệ thống trợ giúp trực tuyến và/ hoặc tài liệu hướng dẫn, so sánh với tổng số nhiệm vụ được kiểm tra.	X = A/B A = Số nhiệm vụ được hoàn thành đúng sau khi truy nhập vào hệ thống trợ giúp trực tuyến B = Tổng số nhiệm vụ được kiểm tra	Càng gần bằng 1.0 càng tốt	Người sử dụng Người bảo trì
Tính hiệu quả của các tài liệu hướng dẫn người sử dụng và/ hoặc hệ thống trợ giúp trong khi sử dụng	Tỷ lệ các chức năng có thể được sử dụng đúng đắn sau khi đọc tài liệu hướng dẫn hoặc sử dụng các hệ thống trợ giúp là bao nhiêu?	Quan sát thái độ của người sử dụng. Đếm số lượng các chức năng được sử dụng đúng đắn sau khi đọc tài liệu hướng dẫn hoặc sử dụng hệ thống trợ giúp và so sánh với tổng số các chức năng.	X = A/B A = Số chức năng có thể được sử dụng B = Tổng số các chức năng được cung cấp	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Khả năng truy cập trợ giúp	Tỷ lệ các chủ đề trợ giúp người sử dụng có thể xác định là bao nhiêu?	Tiến hành kiểm tra người sử dụng và quan sát thái độ của họ. Đếm số nhiệm vụ có hỗ trợ trực tuyến được xác định và so sánh với tổng số các nhiệm vụ được kiểm tra.	X = A/B A = Số nhiệm vụ có trợ giúp trực tuyến được xác định B = Tổng số nhiệm vụ được kiểm tra	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Tần suất trợ giúp	Người sử dụng thường xuyên truy cập vào hệ thống trợ giúp học cách vận hành để hoàn thành công việc của họ như thế nào?	Tiến hành kiểm tra người sử dụng và quan sát thái độ của họ. Đếm số trường hợp người sử dụng truy cập vào hệ thống trợ giúp để hoàn thành công việc của mình.	X = A A = Số lượng truy cập vào hệ thống trợ giúp cho đến khi người sử dụng hoàn thành công việc của mình.	Càng gần bằng 0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng

4.3.3. Khả năng vận hành

Phép đánh giá khả năng dễ vận hành có khả năng ước lượng người sử dụng có thể vận hành và điều khiển phần mềm hay không.

Bảng 11: Bảng các phép đánh giá khả năng dễ vận hành

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Tính phù hợp của vận hành trong sử dụng	Các thành phần giao diện người sử dụng phù hợp như thế nào?	Quan sát thái độ của người sử dụng và hỏi ý kiến của họ.	a) X = 1 - A/B A = Số bản tin hoặc chức năng người sử dụng thấy không phù hợp với mong muốn của mình B = Số bản tin hoặc chức năng b) Y = N/UOT N = Số lần vận hành người sử dụng thấy không phù hợp với mong muốn. UOT = Thời gian vận hành của người sử dụng (trong khoảng thời gian quan sát	Càng gần bằng 1.0 càng tốt Càng gần bằng 0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Sửa lỗi	Người sử dụng có thể dễ dàng sửa lỗi trong các nhiệm vụ không?	Tiến hành kiểm tra người sử dụng và quan sát thái độ của họ.	T = Tc - Ts Tc = Thời điểm hoàn thành việc sửa lỗi nhất định của nhiệm vụ thực thi. Ts = Thời điểm bắt đầu sửa các loại lỗi nhất định của nhiệm vụ được thực thi.	Càng nhỏ càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng

Sửa lỗi trong sử dụng	Người sử dụng có thể dễ dàng khôi phục lỗi hoặc thực hiện lại công việc? Người sử dụng có thể dễ dàng khôi phục lại đầu vào không?	Quan sát thái độ của người sử dụng đang vận hành phần mềm.	a) X = A/UOT A = Số lần người sử dụng hủy bỏ thành công các vận hành lỗi của họ UOT = Thời gian vận hành của người sử dụng trong quá trình quan sát. b) X = A/B A = Số lượng bảng hay biểu mẫu trong đó dữ liệu đầu vào được sửa đổi hay thay thế thành công trước khi được gia công B = Số lượng bảng hay biểu mẫu trong đó người sử dụng cố gắng sửa đổi hay thay thế dữ liệu đầu vào trong thời gian vận hành của người sử dụng được quan sát	Càng lớn càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Tính sẵn sàng của giá trị mặc định trong sử dụng	Người sử dụng có thể dễ dàng chọn các giá trị tham số để thuận tiện trong khi vận hành?	Quan sát thái độ của người sử dụng vận hành phần mềm. Đếm số lần người sử dụng cố gắng thiết lập hoặc lựa chọn các giá trị tham số và thất bại (bởi vì người sử dụng không thể sử dụng các giá trị mặc định cung cấp bởi phần mềm)	X = 1 - A/B A = Số lần người sử dụng thất bại trong việc thiết lập hoặc lựa chọn giá trị tham số trong thời gian ngắn (bởi vì họ không thể sử dụng các giá trị mặc định cung cấp bởi phần mềm) B = Tổng số lần người sử dụng lựa chọn hoặc thiết lập các giá trị tham số	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng

Khả năng hiểu được bản tin trong sử dụng	Người sử dụng có thể dễ dàng hiểu các bản tin từ hệ thống phần mềm? Liệu có bản tin nào gây ra khó hiểu cho người sử dụng trước khi bắt đầu thực hiện thao tác tiếp theo? Người sử dụng có thể dễ dàng nhớ các bản tin quan trọng hay không?	Quan sát thái độ của người sử dụng vận hành phần mềm.	X = A/UOT A = Số lần người sử dụng dừng lại trong thời gian dài hoặc liên tục và lặp lại thất bại thực hiện cùng một thao tác, do không hiểu rõ bản tin. UOT = Thời gian vận hành của người sử dụng (khoảng thời gian quan sát)	Càng gần bằng 0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Các bản tin lỗi rõ ràng	Tỷ lệ lỗi người sử dụng đề xuất hành động phục hồi lỗi chính xác trong điều kiện bị lỗi là bao nhiêu?	Tiến hành kiểm tra người sử dụng và quan sát thái độ của họ.	X = A/B A = Số lượng lỗi mà được người sử dụng đề xuất hành động phục hồi chính xác. B = Số lượng lỗi được kiểm tra	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Khả năng phục hồi lỗi vận hành trong sử dụng	Liệu người sử dụng có thể dễ dàng phục hồi tình trạng xấu nhất của họ không?	Quan sát thái độ của người sử dụng vận hành phần mềm	X = 1 - A/B A = Số trường hợp phục hồi lỗi không thành công (sau khi có lỗi, hoặc thay đổi của người sử dụng) mà người sử dụng không được hệ thống thông báo có rủi ro B = Số lượng lỗi hoặc thay đổi của người sử dụng	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Khoảng thời gian giữa các vận hành lỗi do con người trong sử dụng	Liệu người sử dụng có thể vận hành phần mềm trong thời gian dài mà không có lỗi?	Quan sát thái độ của người sử dụng vận hành phần mềm.	X = T/N (tại thời điểm t trong khoảng [t-T,t]) T = Thời gian vận hành trong quá trình quan sát (hoặc tổng thời gian vận hành giữa các lần vận hành lỗi của người sử dụng) N = Số lần xảy ra lỗi vận hành do con người	Càng lớn càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng

Khả năng tháo gỡ (Sửa lỗi của người sử dụng)	Người sử dụng thường xuyên sửa lỗi đầu vào thành công như thế nào? Người sử dụng thường xuyên hủy lỗi chính xác như thế nào?	Tiến hành kiểm tra người sử dụng và quan sát thái độ của họ.	a) X = A/B A = Số lượng lỗi đầu vào người sử dụng sửa thành công B = Số lượng cố gắng sửa lỗi đầu vào b) Y = A/B A = Số tình trạng lỗi được người sử dụng sửa thành công B = Tổng số tình trạng lỗi được kiểm tra	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Khả năng tùy biến	Người sử dụng có thể dễ dàng tùy biến các thủ tục vận hành cho phù hợp với họ không? Người hướng dẫn người sử dụng có thể dễ dàng thiết lập các mẫu thủ tục vận hành để tránh các lỗi không? Tỷ lệ các chức năng có thể được tùy biến là bao nhiêu?	Tiến hành kiểm tra người sử dụng và quan sát thái độ của họ.	X = A/B A = Số chức năng được tùy biến thành công B = Số lần cố gắng thực hiện tùy biến	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Giảm thủ tục vận hành	Liệu người sử dụng có thể dễ dàng giảm các thủ tục vận hành cho phù hợp với họ không?	Đếm thao tác của người sử dụng cho một quá trình vận hành nhất định, và so sánh giữa thời điểm trước và sau khi tùy biến vận hành.	X = 1 - A/B A = Số thủ tục vận hành được giảm bớt sau khi tùy biến vận hành. B = Số lượng các thủ tục vận hành trước khi tùy biến vận hành	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng
Khả năng truy cập vật lý	Tỷ lệ các chức năng có thể được truy cập bởi tác động vật lý của người sử dụng là bao nhiêu?	Tiến hành kiểm tra người sử dụng và quan sát thái độ của họ.	X = A Người sử dụng Người thiết kế giao diện người sử dụng	Càng gần bằng 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng

4.3.4. Tính hấp dẫn

Phép đánh giá tính hấp dẫn có khả năng đánh giá diện mạo của phần mềm, và sẽ bị tác động bởi các yếu tố như thiết kế và màu của màn hình. Điều này đặc biệt quan trọng đối với các sản phẩm thương mại.

Bảng 12: Bảng các phép đánh giá tính hấp dẫn

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Tương tác với người sử dụng	Giao diện với người sử dụng thân thiện như thế nào?	Bản câu hỏi dành cho người sử dụng	Bản câu hỏi ước định sự thân thiện của giao diện phần mềm với người sử dụng phần mềm, sau quá trình sử dụng.	Phụ thuộc vào phương pháp tính điểm của bản câu hỏi	Người sử dụng Người thiết kế giao diện người sử dụng
Tùy biến thể hiện giao diện	Tỷ lệ các thành phần của giao diện có thể được tùy biến theo sở thích người sử dụng?	Tiến hành kiểm tra người sử dụng và quan sát thái độ.	X = A/B A = Số các phần giao diện có thể tùy biến thỏa mãn người sử dụng. B = Số các phần giao diện người sử dụng muốn tùy biến.	Càng gần tới 1.0 càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng

4.4. Tính hiệu quả

Phép đánh giá tính hiệu quả có khả năng đo các thuộc tính như thời gian tiêu tốn và hoạt động sử dụng tài nguyên của hệ thống máy tính trong quá trình kiểm tra hoặc vận hành. Đối với thời gian hoạt động chia thành 3 loại:

- Thời gian đáp ứng: Thời gian cần thiết nhận được kết quả từ khi nhấn phím truyền. Điều này có nghĩa là thời gian đáp ứng bao gồm cả thời gian xử lý và thời gian truyền tải.

- Thời gian xử lý: Thời gian trôi qua trong máy tính giữa thời điểm nhận bản tin và gửi kết quả. Trong một số trường hợp nó bao gồm thời gian mào đầu vận hành, trong trường hợp khác nó chỉ có nghĩa thời gian sử dụng cho chương trình ứng dụng.

- Thời gian hoàn thành: Thời gian cần thiết để nhận được kết quả từ yêu cầu. Trong nhiều trường hợp thời gian hoàn thành bao gồm nhiều thời gian đáp ứng. Ví dụ, trong trường hợp rút tiền mặt, thời gian hoàn thành là thời gian từ khi ấn phím khởi tạo cho đến khi lấy.

4.4.1. Tiết kiệm thời gian

Phép đánh giá thời gian hoạt động có khả năng đo các thuộc tính như thời gian hoạt động của hệ thống máy tính bao gồm phần mềm trong quá trình kiểm tra hay vận hành.

Bảng 13: Bảng các phép đánh giá thời gian hoạt động

Tên phép đánh giá	Mục đích của phép đánh giá	Phương pháp áp dụng	Phép đo, công thức và tính toán các thành phần dữ liệu	Chuyển đổi giá trị đo	Đối tượng sử dụng
Thời gian đáp ứng	Thời gian cần để hoàn thành một nhiệm vụ nhất định là bao nhiêu? Thời gian cần thiết trước khi hệ thống trả lời một vận hành nhất định là bao nhiêu?	Bắt đầu một nhiệm vụ nhất định. Đo thời gian ví dụ cần thiết để hoàn thành nhiệm vụ đó. Lưu kết quả cho mỗi lần đo	T = (thời điểm nhận được kết quả đo) - (thời điểm kết thúc vào lệnh)	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Thời gian đáp ứng (Thời gian đáp ứng trung bình)	Thời gian đợi trung bình của người sử dụng từ khi đưa ra yêu cầu đến khi yêu cầu được hoàn thành trong điều kiện tải hệ thống nhất định trên các nhiệm vụ đồng thời và tải sử dụng hệ thống là bao nhiêu?	Thực hiện một số kịch bản các nhiệm vụ đồng thời Đo thời gian cần thiết hoàn thành vận hành lựa chọn. Lưu kết quả của mỗi lần đo và tính giá trị trung bình cho mỗi kịch bản.	X = Tmean/TXmean Tmean = ∑(Ti)/N, (với i = 1 đến N) TXmean = Thời gian đáp ứng trung bình yêu cầu Ti = Thời gian đáp ứng của lần đánh giá thứ i N = Số lần đánh giá	Càng xấp xỉ bằng 1 càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Thời gian đáp ứng (Tỷ lệ thời gian đáp ứng trong trường hợp xấu nhất)	Giới hạn tuyệt đối về thời gian cần thiết để hoàn thành một chức năng là bao nhiêu? Trong trường hợp xấu nhất, liệu người sử dụng vẫn có thể nhận được phản hồi trong giới hạn thời gian nhất định không? Trong trường hợp xấu nhất, liệu người sử dụng vẫn nhận được trả lời từ phần mềm trong thời gian đủ ngắn người sử dụng chấp nhận được không?	Điều chỉnh bài kiểm tra. Giả lập điều kiện để hệ thống đạt được một trạng thái tải lớn nhất. Thực hiện chương trình và giám sát kết quả.	X = Tmax/Rmax Tmax = MAX(Ti) (với i = 1 đến N) Rmax = Thời gian đáp ứng lớn nhất yêu cầu MAX(Ti) = Thời gian đáp ứng lớn nhất trong các lần đánh giá N = Số lần đánh giá Ti = Thời gian đáp ứng lần đánh giá i	Càng gần bằng 1.0 và nhỏ hơn 1.0 càng tốt	Người sử dụng Người phát triển Người bảo trì SQA

Thông lượng	Bao nhiêu nhiệm vụ có thể được thực hiện thành công trong khoảng thời gian cho trước?	Điều chỉnh mỗi nhiệm vụ tương ứng với mức độ ưu tiên định trước. Bắt đầu một số các nhiệm vụ. Đo thời gian cần thiết để nhiệm vụ được đo hoàn thành hoạt động của nó. Lưu lại kết quả cho mỗi lần đo.	X = A/T A = Số nhiệm vụ được hoàn thành T = Khoảng thời gian quan sát	Càng lớn càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Thông lượng (Giá trị trung bình của thông lượng)	Số nhiệm vụ trung bình đồng thời hệ thống có thể xử lý được trong đơn vị thời gian thiết lập là bao nhiêu?	Điều chỉnh mỗi nhiệm vụ tương ứng với mức độ ưu tiên định trước. Thực hiện một số các nhiệm vụ đồng thời. Đo thời gian cần thiết để hoàn thành nhiệm vụ được chọn trên điều kiện lưu lượng cho trước. Lưu lại kết quả cho mỗi lần đo.	X = Xmean/Rmean Xmean = ∑(Xi) / N Rmean = Thông lượng trung bình yêu cầu Xi = Ai/Ti Ai = Số nhiệm vụ đồng thời quan sát trong khoảng thời gian thiết lập cho lần đánh giá thứ i Ti = Khoảng thời gian thiết lập cho lần đánh giá thứ i N = Số lần đánh giá	Càng lớn càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Thông lượng (Tỷ lệ thông lượng trong trường hợp xấu nhất)	Giới hạn tuyệt đối trong hệ thống trên số lượng và xử lý các nhiệm vụ đồng thời như là thông lượng là bao nhiêu?	Điều chỉnh bài kiểm tra. Giả lập điều kiện để hệ thống đạt được trạng thái tải lớn nhất. Thực hiện các nhiệm vụ đồng thời và giám sát kết quả.	X = Xmax/Rmax Xmax = MAX(Xi) (với i = 1 đến N) Rmax = Thông lượng lớn nhất yêu cầu MAX(Xi) = Số lớn nhất các nhiệm vụ trong các lần đánh giá Xi = Ai/Ti Ai = Số nhiệm vụ đồng thời quan sát trong khoảng thời gian thiết lập cho lần đánh giá thứ i Ti = Khoảng thời gian thiết lập cho lần đánh giá thứ i N = Số lần đánh giá	Càng lớn càng tốt	Người sử dụng Người phát triển Người bảo trì SQA

Thời gian hoàn thành	Thời gian đợi của người sử dụng sau khi đưa lệnh để bắt đầu một nhóm các nhiệm vụ liên quan và khi hoàn thành chúng là bao nhiêu?	Điều chỉnh bài kiểm tra tương ứng. Bắt đầu thực hiện nhiệm vụ. Đo thời gian cần thiết cho nhiệm vụ để hoàn thành vận hành. Lưu kết quả đo mỗi lần đo.	T = Thời gian giữa kết thúc nhận được kết quả đầu ra và kết thúc đưa yêu cầu của người sử dụng.	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Thời gian hoàn thành (Thời gian trung bình cho việc hoàn thành)	Thời gian đợi trung bình của người sử dụng sau khi đưa lệnh để bắt đầu một nhóm các nhiệm vụ liên quan và khi hoàn thành chúng với tải hệ thống xác định trên các nhiệm vụ đồng thời và tải sử dụng hệ thống là bao nhiêu?	Điều chỉnh bài kiểm tra tương ứng. Giả lập điều kiện tải của hệ thống bằng cách thực hiện một số các nhiệm vụ đồng thời. Đo thời gian cần thiết để hoàn thành nhiệm vụ đã chọn trên lưu lượng cho trước. Lưu kết quả cho mỗi lần đo.	X = Tmean/TXmean Tmean = ∑(Ti)/N, (với i + 1 đến N) TXmean = Thời gian hoàn thành trung bình yêu cầu Ti = Thời gian hoàn thành cho lần đánh giá thứ i N = Số lần đánh giá	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Thời gian hoàn thành (Tỷ lệ thời gian hoàn thành trong trường hợp xấu nhất)	Giới hạn tuyệt đối về thời gian yêu cầu để hoàn thành nhiệm vụ là bao nhiêu? Trong trường hợp xấu nhất thời gian để hệ thống phần mềm hoàn thành các nhiệm vụ xác định là bao nhiêu?	Điều chỉnh bài kiểm tra. Giả lập điều kiện để hệ thống đạt được trạng thái tải lớn nhất trên các nhiệm vụ thực hiện. Thực hiện các nhiệm vụ được chọn và giám sát kết quả.	X = Tmax/Rmax Tmax = MAX(Ti) (với i = 1 đến N) Rmax = Thời gian hoàn thành lớn nhất yêu cầu MAX(Ti) = Thời gian hoàn thành lớn nhất trong các lần đánh giá N = Số lần đánh giá Ti = Thời gian hoàn thành cho lần đánh giá thứ i	Càng gần 1.0 và nhỏ hơn 1.0 càng tốt	Người sử dụng Người phát triển Người bảo trì SQA

Thời gian đợi	Tỷ lệ thời gian người sử dụng đợi hệ thống trả lời là bao nhiêu?	Thực hiện một số các kịch bản nhiệm vụ đồng thời. Đo thời gian cần thiết để hoàn thành các vận hành đã chọn. Lưu kết quả của mỗi lần thử nghiệm và tính thời gian trung bình cho mỗi kịch bản.	X = Ta/Tb Ta = Tổng thời gian đợi Tb = thời gian của nhiệm vụ	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA

4.4.2. Sử dụng tài nguyên

Phép đánh giá sử dụng tài nguyên có khả năng đo các thuộc tính như hoạt động của tài nguyên sử dụng của hệ thống máy tính bao gồm phần mềm trong quá trình kiểm tra hay vận hành thực tế.

Bảng 14: Bảng các phép đánh giá sử dụng tài nguyên

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Sử dụng thiết bị I/O	Việc sử dụng thiết bị I/O có quá cao, gây ra không hiệu quả?	Thực hiện đồng thời một lượng lớn các nhiệm vụ, ghi lại sử dụng của thiết bị I/O, và so sánh với mục tiêu thiết kế.	X = A/B A = Thời gian thiết bị I/O được sử dụng. B = thời gian xác định được thiết kế cho việc sử dụng thiết bị.	Nhỏ hơn và càng gần 1.0 thì tốt hơn	Người phát triển Người bảo trì SQA
Giới hạn tải I/O	Giới hạn tuyệt đối của việc sử dụng I/O trong hoàn thành chức năng?	Điều chỉnh điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải lớn nhất. Thực hiện ứng dụng và giám sát kết quả.	X = Amax/Rmax Amax = MAX(Ai), (với i = 1 đến N) Rmax = Bản tin I/O lớn nhất yêu cầu MAX(Ai) = Số lượng lớn nhất bản tin I/O từ lần 1 đến lần đánh giá thứ i. N = Số lần đánh giá	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA

Lỗi liên quan đến I/O	Người sử dụng thường xuyên gặp vấn đề trong vận hành liên quan đến thiết bị I/O như thế nào?	Điều chỉnh điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải lớn nhất. Thực hiện ứng dụng và ghi lại số lỗi do sự cố I/O và các cảnh báo.	X = A/T A = Số bản tin cảnh báo hoặc sự cố hệ thống T = Thời gian vận hành người sử dụng trong quá trình quan sát.	Càng nhỏ càng tốt	Người sử dụng Người bảo trì SQA
Tỷ lệ hoàn thành I/O trung bình	Số lượng trung bình của bản tin lỗi liên quan đến I/O và số lần sự cố trong một thời gian xác định và sử dụng xác định là bao nhiêu?	Điều chỉnh điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải lớn nhất. Thực hiện ứng dụng và ghi lại số lỗi do sự cố I/O và các cảnh báo.	X = Amean/Rmean Amean = ∑(Ai)/N Rmean = Số bản tin I/O trung bình được yêu cầu Ai = Số bản tin lỗi cho lần đánh giá thứ i N = Số lần đánh giá	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Thời gian đợi của người sử dụng khi sử dụng thiết bị I/O	Tác động của việc sử dụng thiết bị I/O tới thời gian đợi của người sử dụng là bao nhiêu?	Thực hiện đồng thời một lượng lớn các nhiệm vụ và đo thời gian đợi vận hành của thiết bị I/O của người sử dụng.	T = Thời gian đợi kết thúc vận hành thiết bị I/O	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Sử dụng bộ nhớ lớn nhất	Giới hạn tuyệt đối của bộ nhớ được yêu cầu để hoàn thành chức năng là bao nhiêu?	Điều chỉnh điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải lớn nhất. Thực hiện ứng dụng và giám sát kết quả	X = Amax/Rmax Amax = MAX(Ai), (với i = 1 đến N) Rmax = Số lượng bản tin lỗi lớn nhất liên quan đến bộ nhớ được yêu cầu MAX(Ai) = Số lượng bản tin lỗi lớn nhất liên quan đến bộ nhớ từ lần đánh giá 1 đến lần i. N = Tổng số lần đánh giá	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA

Xuất hiện trung bình lỗi bộ nhớ	Số lượng trung bình của bản tin lỗi và sự cố liên quan đến bộ nhớ trong một thời gian xác định và tải xác định trong hệ thống là bao nhiêu?	Điều chỉnh điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải lớn nhất. Thực hiện ứng dụng và ghi lại số lỗi do sự cố bộ nhớ và cảnh báo.	X = Amean/Rmean Amean = ∑(Ai)/N Rmean = Số bản tin lỗi trung bình liên quan đến bộ nhớ được yêu cầu Ai = Số bản tin lỗi liên quan đến bộ nhớ trong lần đánh giá thứ i N = Số lần đánh giá	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Tỷ lệ lỗi bộ nhớ/ thời gian	Số lượng lỗi bộ nhớ trong khoảng thời gian thiết lập và sử dụng tài nguyên nhất định là bao nhiêu?	Điều chỉnh các điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải lớn nhất. Thực hiện ứng dụng và ghi lại số lỗi do sự cố bộ nhớ và cảnh báo.	X = A/T A = Số lượng bản tin cảnh báo, sự cố hệ thống T = Thời gian vận hành người sử dụng trong quá trình quan sát.	Càng nhỏ càng tốt	Người sử dụng Người bảo trì SQA
Sử dụng truyền dẫn lớn nhất	Giới hạn tuyệt đối của truyền dẫn yêu cầu để hoàn thành chức năng là bao nhiêu?	Xác định yêu cầu để hệ thống đạt được trạng thái tải lớn nhất. Giả lập điều kiện đó. Thực hiện ứng dụng và giám sát kết quả.	X = Amax/Rmax Amax = MAX(Ai), (với i = 1 đến N) Rmax = Số lớn nhất được yêu cầu của bản tin lỗi và sự cố liên quan đến truyền dẫn MAX(Ai) = Số lớn nhất của bản tin lỗi và sự cố liên quan đến truyền tải từ lần đánh giá từ 1 đến lần i. N = Số lần đánh giá	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Cân bằng sử dụng giữa các thiết bị phương tiện	Mức đồng bộ giữa các phương tiện khác nhau trong một khoảng thời gian thiết lập là thế nào?	Điều chỉnh các điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải truyền dẫn lớn nhất. Thực hiện ứng dụng và ghi lại độ trễ trong quá trình xử lý các loại phương tiện khác nhau.	X = SyncTime/T SyncTime = Thời gian dành cho tài nguyên liên tục T = Khoảng thời gian được yêu cầu trong đó các phương tiện không giống nhau được mong đợi kết thúc các nhiệm vụ một cách đồng bộ	Càng nhỏ càng tốt	Người sử dụng Người bảo trì SQA

Xuất hiện trung bình lỗi truyền dẫn	Số lượng trung bình bản tin lỗi và sự cố liên quan đến truyền tải trong khoảng thời gian xác định và sử dụng xác định là bao nhiêu?	Điều chỉnh điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải lớn nhất. Thực hiện ứng dụng và ghi lại số lỗi do sự cố truyền tải và các cảnh báo.	X = Amean/Rmean Amean = ∑(Ai)/N Rmean = Số lượng trung bình được yêu cầu các bản tin lỗi và sự cố liên quan đến truyền dẫn Ai = Số lượng trung bình các bản tin lỗi và sự cố liên quan đến truyền dẫn trong lần đánh giá thứ i. N = Số lần đánh giá	Càng nhỏ càng tốt	Người sử dụng Người phát triển Người bảo trì SQA
Lỗi truyền dẫn trung bình trên thời gian	Có bao nhiêu bản tin lỗi liên quan đến truyền dẫn xảy ra trong một khoảng thời gian xác định và sử dụng tài nguyên xác định?	Điều chỉnh các điều kiện kiểm tra. Giả lập điều kiện hệ thống đạt được trạng thái tải truyền dẫn lớn nhất. Thực hiện ứng dụng và ghi lại số lỗi do sự cố truyền dẫn và các cảnh báo.	X = A/T A = Số lượng bản tin cảnh báo hay sự cố hệ thống T = Thời gian vận hành người sử dụng trong quá trình quan sát.	Càng nhỏ càng tốt	Người sử dụng Người bảo trì SQA
Sử dụng dung lượng truyền dẫn	Hệ thống phần mềm có khả năng thực hiện các nhiệm vụ trong phạm vi dung lượng truyền dẫn mong đợi được không?	Thực hiện đồng thời các nhiệm vụ với nhiều người sử dụng, quan sát dung lượng truyền dẫn và so sánh giá trị.	X = A/B A = dung lượng truyền dẫn B = dung lượng truyền dẫn xác định được thiết kế để phần mềm sử dụng thực hiện công việc.	Nhỏ hơn và càng gần 1.0 thì càng tốt	Người sử dụng Người bảo trì SQA

4.5. Khả năng bảo trì

Phép đánh giá khả năng bảo trì có khả năng đo các thuộc tính như hoạt động của người bảo trì, người sử dụng, hoặc hệ thống chứa phần mềm, khi phần mềm được bảo trì hoặc thay đổi trong quá trình kiểm tra hoặc bảo trì.

4.5.1. Khả năng phân tích

Phép đánh giá khả năng phân tích có khả năng đo các thuộc tính cố gắng của người bảo trì hay người sử dụng hay tiêu tốn tài nguyên khi cố gắng chẩn đoán thiếu sót hoặc nguyên nhân của sự cố, hoặc xác định các phần bị thay đổi.

Bảng 15: Bảng các phép đánh giá khả năng phân tích

Tên phép đánh giá	Mục đích của phép đánh giá	Phương pháp áp dụng	Phép đo, công thức và tính toán các thành phần dữ liệu	Chuyển đổi giá trị đo	Đối tượng sử dụng
Khả năng theo dõi kiểm tra	Người sử dụng có thể nhận biết vận hành cụ thể nào gây ra sự cố không? Người bảo trì có thể dễ dàng tìm vận hành cụ thể nào gây ra sự cố không?	Quan sát thái độ người sử dụng hoặc người bảo trì đang cố gắng xử lý sự cố	X = A/B A = Số lượng dữ liệu thực tế được ghi trong quá trình vận hành. B = Số lượng dữ liệu dự kiến được ghi đủ để giám sát trạng thái phần mềm trong quá trình vận hành	Càng gần 1.0 càng tốt	Người phát triển Người bảo trì Người vận hành
Hỗ trợ chức năng chẩn đoán	Các chức năng chẩn đoán hỗ trợ phân tích nguyên nhân có khả năng đến đâu? Người sử dụng có thể nhận biết vận hành cụ thể nào gây ra sự cố không? (Người sử dụng có thể có khả năng tránh rơi vào sự cố giống như vậy với vận hành khác) Người bảo trì có thể dễ dàng tìm nguyên nhân sự cố hay không?	Quan sát thái độ người sử dụng hoặc người bảo trì đang cố gắng xử lý sự cố sử dụng các chức năng chẩn đoán	X = A/B A = Số sự cố người bảo trì có thể chẩn đoán (sử dụng chức năng chẩn đoán) để hiểu mối quan hệ nguyên nhân - kết quả B = Tổng số sự cố ghi được	Càng gần 1.0 càng tốt	Người phát triển Người bảo trì Người vận hành
Khả năng phân tích lỗi	Người sử dụng có thể nhận biết vận hành cụ thể nào gây ra sự cố không? Người bảo trì có thể dễ dàng tìm ra nguyên nhân sự cố không?	Quan sát thái độ người sử dụng hoặc người bảo trì đang cố gắng xử lý sự cố	X = 1 - A/B A = Số sự cố nguyên nhân còn chưa tìm được B = Tổng số sự cố ghi được	Càng xấp xỉ bằng 1 càng tốt	Người sử dụng Người phát triển Người bảo trì Người vận hành

Hiệu quả phân tích sự cố	Người sử dụng có thể phân tích hiệu quả nguyên nhân sự cố không? (Người sử dụng thỉnh thoảng thực hiện bảo trì bằng cách thiết lập các thông số) Người bảo trì có thể dễ dàng tìm ra nguyên nhân sự cố không? Việc phân tích nguyên nhân sự cố dễ dàng đến mức nào?	Quan sát thái độ người sử dụng hoặc người bảo trì đang cố gắng xử lý sự cố	X = Sum(T)/N T = Tout - Tin Tout = Thời điểm các nguyên nhân sự cố được phát hiện (hoặc được thông báo cho người sử dụng) Tin = Thời điểm nhận được thông báo lỗi N = Số lỗi ghi được	Càng nhỏ càng tốt	Người phát triển Người bảo trì Người vận hành
Khả năng giám sát trạng thái	Người sử dụng có thể nhận biết vận hành cụ thể gây ra sự cố thông qua dữ liệu giám sát trong quá trình vận hành không? Người bảo trì có thể dễ dàng tìm ra nguyên nhân sự cố thông qua dữ liệu giám sát trong quá trình vận hành không?	Quan sát thái độ người sử dụng hoặc người bảo trì đang cố gắng lấy thông tin giám sát ghi trạng thái phần mềm trong quá trình vận hành.	X = 1 - A/B A = Số trường hợp người bảo trì (hoặc người sử dụng) không lấy được dữ liệu giám sát B = Số lượng trường hợp mà người bảo trì (người sử dụng) cố gắng lấy dữ liệu giám sát ghi trạng thái phần mềm trong quá trình vận hành	Càng gần bằng 1.0 càng tốt	Người sử dụng Người phát triển Người bảo trì Người vận hành

4.5.2. Khả năng thay đổi

Phép đánh giá khả năng thay đổi được có khả năng đo các thuộc tính như sự cố gắng của người bảo trì hay người sử dụng bằng cách đo hoạt động của người bảo trì, người sử dụng, hay hệ thống chứa phần mềm khi thử triển khai thay đổi xác định.

Bảng 16: Bảng các phép đánh giá khả năng thay đổi

Tên phép đánh giá	Mục đích của phép đánh giá	Phương pháp áp dụng	Phép đo, công thức và tính toán các thành phần dữ liệu	Chuyển đổi giá trị đo	Đối tượng sử dụng
Tính hiệu quả chu trình thay đổi	Vấn đề của người sử dụng có thể được giải quyết thỏa đáng trong khoảng thời gian hợp lý không?	Giám sát tương tác giữa người sử dụng và nhà cung cấp. Ghi lại thời gian từ khi người sử dụng bắt đầu yêu cầu đến khi vấn đề được giải quyết.	Thời gian trung bình: Tav = Sum(Tu)/N Tu = Trc - Tsn Tsn = Thời điểm người sử dụng hoàn thành gửi yêu cầu để bảo trì cho nhà cung cấp cùng báo cáo vấn đề Trc = Thời điểm người sử dụng nhận được phiên bản sửa lại (hay báo cáo trạng thái) N = Số phiên bản	Càng nhỏ càng tốt, trừ phi số phiên bản sửa lại lớn	Người sử dụng Người bảo trì Người vận hành
Thời gian triển khai thay đổi phần mềm	Liệu người bảo trì có thể dễ dàng thay đổi phần mềm để giải quyết sự cố không?	Quan sát thái độ của người sử dụng và người bảo trì khi cố gắng thay đổi phần mềm. Nếu không xem xét báo cáo giải quyết vấn đề hay báo cáo bảo trì	Thời gian trung bình: Tav = Sum(Tm)/N Tm = Tout - Tin Tout = Thời điểm nguyên nhân sự cố được gỡ bỏ bằng cách thay đổi phần mềm (hoặc trạng thái được báo cáo cho người sử dụng) Tin = Thời điểm nguyên nhân sự cố được phát hiện N = Số sự cố được ghi lại và được gỡ bỏ	Càng nhỏ càng tốt, trừ phi số sự cố lớn	Người sử dụng Người bảo trì Người vận hành
Tính phức tạp của qsuá trình sửa đổi	Người bảo trì có thể dễ dàng thay đổi phần mềm để giải quyết vấn đề không?	Quan sát thái độ người bảo trì trong đang cố gắng thay đổi phần mềm. Nếu không xem xét báo cáo giải quyết lỗi hay báo cáo bảo trì và mô tả sản phẩm	T = Sum(AB)/N A = Thời gian làm việc để thay đổi B = Khối lượng thay đổi phần mềm N = Số lượng các thay đổi	Càng nhỏ càng tốt, hay số lượng yêu cầu thay đổi quá nhiều	Người sử dụng Người bảo trì Người vận hành

Khả năng thay đổi tham số	Người sử dụng hay người bảo trì có thể dễ dàng thay đổi tham số để thay đổi phần mềm và giải quyết vấn đề không?	Quan sát thái độ của người sử dụng hoặc người bảo trì khi cố gắng thay đổi phần mềm. Nếu không xem xét báo cáo giải quyết vấn đề hay báo cáo bảo trì	X = 1 - A/B A = Số trường hợp người bảo trì thất bại khi thay đổi phần mềm bằng cách sử dụng tham số B = Số trường hợp người bảo trì cố gắng thay đổi phần mềm bằng cách sử dụng tham số	Càng gần đến 1.0 càng tốt	Người sử dụng Người bảo trì Người vận hành
Khả năng quản lý thay đổi phần mềm	Người sử dụng có thể dễ dàng nhận biết phiên bản sửa lại không? Người bảo trì có thể dễ dàng thay đổi phần mềm để giải quyết vấn đề không?	Quan sát thái độ người sử dụng hoặc người bảo trì đang cố gắng thay đổi phần mềm. Nếu không xem xét báo cáo giải quyết vấn đề hay báo cáo bảo trì	X = A/B A = Số lượng dữ liệu thay đổi được ghi lại thực tế B = Số lượng dữ liệu thay đổi dự kiến được ghi vừa đủ để dò tìm thay đổi phần mềm.	Càng gần đến một càng tốt hay gần đến 0 càng thay đổi ít	Người sử dụng Người bảo trì Người vận hành

4.5.3. Tính cân bằng (ổn định)

Phép đánh giá tính ổn định có khả năng đo các thuộc tính liên quan đến hoạt động không mong đợi của hệ thống chứa phần mềm khi phần mềm được kiểm tra hoặc vận hành sau khi thay đổi.

Bảng 17: Bảng các phép đánh giá tính ổn định

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Tỷ lệ thay đổi thành công	Người sử dụng có thể vận hành hệ thống phần mềm không sự cố sau khi được bảo trì không? Người bảo trì có thể dễ dàng giảm sự cố gây ra bởi tác động của việc bảo trì không?	Quan sát thái độ của người sử dụng hay người bảo trì đang vận hành hệ thống phần mềm sau bảo trì. Đếm số lượng sự cố mà người sử dụng hay người bảo trì gặp phải khi vận hành phần mềm trước và sau khi bảo trì Nếu không xem xét báo cáo giải quyết sự cố, báo cáo vận hành hay báo cáo bảo trì	X = Na/Ta Y{(Na/Ta)/(Nb/Tb)} Na = Số trường hợp người sử dụng gặp sự cố sau trong vận hành sau khi phần mềm được thay đổi Nb = Số trường hợp người sử dụng gặp sự cố trong vận hành trước khi phần mềm được thay đổi	Càng nhỏ và càng gần đến 0 càng tốt	Người sử dụng Người bảo trì Người vận hành

			Ta = Thời gian vận hành trong khoảng thời gian quan sát xác định sau khi phần mềm được thay đổi. Tb = Thời gian vận hành trong khoảng thời gian quan sát xác định trước khi phần mềm được thay đổi.		
Xác định ảnh hưởng của thay đổi (Sự cố nảy sinh sau khi thay đổi)	Người sử dụng có thể vận hành hệ thống phần mềm không sự cố sau khi được bảo trì không? Người bảo trì có thể dễ dàng giảm sự cố gây ra bởi tác động của việc bảo trì không?	Đếm sự cố xảy ra sau khi thay đổi, chúng móc xích lẫn nhau và ảnh hưởng bởi thay đổi.	X = A/N A = Số sự cố nảy sinh sau khi sự cố được giải quyết bằng thay đổi trong một thời gian nhất định N = Số sự cố được giải quyết	Càng nhỏ, càng gần đến 0 càng tốt	Người sử dụng Người bảo trì Người vận hành

4.5.4. Khả năng kiểm định (khả năng kiểm tra)

Phép đánh giá khả năng kiểm tra có khả năng đo các thuộc tính như sự cố gắng của người bảo trì hay người sử dụng bằng cách đo hoạt động của người bảo trì, người sử dụng, hay hệ thống chứa phần mềm khi thử kiểm tra phần mềm đã thay đổi hay chưa thay đổi.

Bảng 18: Bảng các phép đánh giá khả năng kiểm tra

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Tính sẵn sàng của chức năng kiểm tra có sẵn sàng trong phần mềm	Liệu người sử dụng và người bảo trì có thể dễ dàng thực hiện kiểm tra vận hành mà không cần chuẩn bị thêm phương tiện kiểm tra không?	Quan sát thái độ của người sử dụng hay người bảo trì kiểm tra hệ thống phần mềm sau khi bảo trì.	X = A/B A = Số trường hợp người bảo trì có thể sử dụng chức năng kiểm tra phù hợp có sẵn B = Số trường hợp thực hiện kiểm tra	Càng gần 1.0 càng tốt.	Người sử dụng Người bảo trì Người vận hành
Tính hiệu quả khi kiểm tra lại	Liệu người sử dụng và người bảo trì có thể dễ dàng kiểm tra vận hành và xác định phần mềm đã sẵn sàng hoạt động hay không?	Quan sát thái độ của người sử dụng hay người bảo trì kiểm tra hệ thống phần mềm sau khi bảo trì.	X = Sum(T)/N T = Thời gian cần thiết để kiểm tra chắc chắn rằng sự cố được báo cáo đã được giải quyết hay chưa N = Số sự cố được giải quyết	Càng nhỏ càng tốt	Người sử dụng Người bảo trì Người vận hành

Khả năng khởi tạo lại việc kiểm tra	Người sử dụng và người bảo trì có thể dễ dàng thực hiện kiểm tra vận hành với các điểm kiểm tra sau khi bảo trì không?	Quan sát thái độ của người sử dụng hay người bảo trì kiểm tra hệ thống phần mềm sau bảo trì.	X = A/B A = Số trường hợp người bảo trì có thể tạm dừng và khởi tạo lại quá trình kiểm tra đang thực hiện tại một số điểm mong muốn để kiểm tra lần lượt từng bước một. B = Số trường hợp tạm dừng quá trình kiểm tra	Càng gần 1.0 càng tốt	Người sử dụng Người bảo trì Người vận hành

4.6. Tính khả chuyển

Phép đánh giá tính khả chuyển có khả năng đo các thuộc tính như hoạt động của người vận hành hay hệ thống trong phạm vi các hoạt động chuyển đổi.

4.6.1. Khả năng tương hợp (khả năng tương thích)

Phép đánh giá khả năng tương thích có khả năng đo các thuộc tính như hoạt động của hệ thống hay người sử dụng đang thử tương hợp phần mềm với các môi trường xác định khác nhau. Khi người sử dụng phải áp dụng các thủ tục tương hợp khác với các thủ tục cung cấp bởi phần mềm trước kia cho các nhu cầu tương hợp cụ thể, nỗ lực của người sử dụng được yêu cầu cho quá trình tương hợp phải được đo.

Bảng 19: Bảng các phép đánh giá khả năng tương thích

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Khả năng tương thích của cấu trúc dữ liệu	Người sử dụng hoặc người bảo trì có thể dễ dàng tương hợp phần mềm với tập dữ liệu trong môi trường mới không?	Quan sát thái độ người sử dụng hoặc người bảo trì khi cố gắng tương hợp phần mềm với môi trường hoạt động mới	X = A/B A = Số lượng dữ liệu có khả năng vận hành nhưng không được nhận biết do vận hành không đầy đủ gây ra bởi các hạn chế trong quá trình tương hợp B = Số lượng dữ liệu được mong đợi có khả năng vận hành trong môi trường tương hợp với phần mềm	Càng gần 1.0 càng tốt.	Người phát triển Người bảo trì Người vận hành

Khả năng tương thích môi trường phần cứng (Khả năng tương thích với các thiết bị phần cứng và thiết bị mạng)	Người sử dụng hoặc người bảo trì có thể dễ dàng tương hợp phần mềm với môi trường không? Hệ thống phần mềm có đủ khả năng tự tương thích với môi trường hoạt động?	Quan sát thái độ của người sử dụng hoặc người bảo trì khi cố gắng tương hợp phần mềm với môi trường hoạt động	X = 1 - A/B A = Số lượng chức năng vận hành các nhiệm vụ của chúng không được hoàn thành hoặc không đạt được kết quả đáp ứng mức độ thỏa đáng trong quá trình kiểm tra vận hành với môi trường công việc người sử dụng B = Tổng số các chức năng được kiểm tra.	Càng lớn càng tốt	Người phát triển Người bảo trì Người vận hành
Khả năng tương thích với môi trường của tổ chức (Khả năng tương thích của tổ chức với hạ tầng của tổ chức)	Người sử dụng hoặc người bảo trì có thể dễ dàng tương hợp phần mềm với môi trường ? Hệ thống phần mềm có đủ khả năng tự tương thích với môi trường hoạt động?	Quan sát thái độ của người sử dụng hoặc người bảo trì khi cố gắng tương hợp phần mềm với môi trường hoạt động	X = 1 - A/B A = Số lượng chức năng vận hành các nhiệm vụ của chúng không được hoàn thành hoặc không đạt được kết quả đáp ứng mức độ thỏa đáng trong quá trình kiểm tra vận hành với môi trường công việc người sử dụng B = Tổng số các chức năng được kiểm tra.	Càng lớn càng tốt	Người phát triển Người bảo trì Người vận hành
Tính thân thiện với người sử dụng	Người sử dụng hoặc người bảo trì có thể dễ dàng tương hợp phần mềm với môi trường?	Quan sát thái độ người sử dụng hoặc người bảo trì khi cố gắng tương hợp phần mềm với môi trường hoạt động	T = Tổng thời gian vận hành người sử dụng để hoàn thành tương thích phần mềm với môi trường người sử dụng khi họ cài đặt hoặc thay đổi cài đặt.	Càng nhỏ càng tốt	Người phát triển Người bảo trì Người vận hành

Khả năng tương thích với môi trường phần mềm hệ thống (Khả năng tương thích với hệ điều hành, phần mềm mạng và phần mềm ứng dụng liên kết)	Người sử dụng hoặc người bảo trì có thể dễ dàng tương hợp phần mềm với môi trường ? Hệ thống phần mềm có đủ khả năng tự tương thích với môi trường hoạt động?	Quan sát thái độ của người sử dụng hoặc người bảo trì khi cố gắng tương hợp phần mềm với môi trường hoạt động	X = 1 - A/B A = Số lượng chức năng vận hành có các nhiệm vụ không được hoàn thành hoặc không đạt được kết quả đáp ứng mức thỏa đáng trong quá trình kiểm tra vận hành phối hợp với phần mềm hệ thống điều hành hoặc phần mềm ứng dụng hiện thời. B = Tổng số các chức năng được kiểm tra.	Càng lớn càng tốt	Người phát triển Người bảo trì Người vận hành

4.6.2. Khả năng cài đặt

Phép đánh giá khả năng cài đặt có khả năng đo các thuộc tính như hoạt động của hệ thống hoặc người sử dụng đang thử cài đặt phần mềm trong môi trường cụ thể người sử dụng.

Bảng 20: Bảng các phép đánh giá khả năng cài đặt

Tên phép đánh giá	**Mục đích của phép đánh giá**	**Phương pháp áp dụng**	**Phép đo, công thức và tính toán các thành phần dữ liệu**	**Chuyển đổi giá trị đo**	**Đối tượng sử dụng**
Dễ dàng cài đặt	Người sử dụng hay người vận hành có thể dễ dàng cài đặt phần mềm trong môi trường hoạt động không?	Quan sát thái độ của người sử dụng hay người bảo trì khi cố gắng cài đặt phần mềm trong môi trường hoạt động	X = A/B A = Số trường hợp người sử dụng thay đổi cài đặt thành công để thuận tiện cho mình B = Tổng số trường hợp người sử dụng cố gắng thay đổi cài đặt để thuận tiện cho mình	Càng gần 1.0 càng tốt.	Người phát triển Người bảo trì Người vận hành
Dễ dàng thực hiện lại cài đặt	Người sử dụng hay người vận hành có thể dễ dàng cài đặt lại phần mềm không?	Quan sát thái độ của người sử dụng hay người bảo trì khi thử cài đặt lại phần mềm	X = 1 - A/B A = Số trường hợp người sử dụng không cài đặt lại được trong quá trình cài đặt B = Tổng số trường hợp người sử dụng cố gắng cài đặt lại trong quá trình cài đặt	Càng gần 1.0 càng tốt.	Người phát triển Người bảo trì Người vận hành

4.6.3. Khả năng chung sống (Khả năng cùng tồn tại)

Phép đánh giá khả năng cùng tồn tại có khả năng đo các thuộc tính như hoạt động của hệ thống hay người sử dụng đang thử sử dụng phần mềm cùng với các phần mềm độc lập khác trong môi trường chung chia sẻ các tài nguyên chung.

Bảng 21: Bảng các phép đánh giá khả năng cùng tồn tại

Tên phép đánh giá	Mục đích của phép đánh giá	Phương pháp áp dụng	Phép đo, công thức và tính toán các thành phần dữ liệu	Chuyển đổi giá trị đo	Đối tượng sử dụng
Sẵn sàng cùng tồn tại	Người sử dụng có thường xuyên gặp phải các ràng buộc hoặc sự cố không mong muốn khi vận hành đồng thời với phần mềm khác?	Sử dụng phần mềm đã được đánh giá đồng thời với các phần mềm khác mà người sử dụng hay dùng	X = A/T A = Số ràng buộc hoặc sự cố không mong muốn người sử dụng gặp phải khi vận hành đồng thời với phần mềm khác T = Thời gian vận hành đồng thời với phần mềm khác	Càng gần 0 càng tốt.	Người phát triển Người bảo trì Người vận hành SQA

4.6.4. Khả năng thay thế

Phép đánh giá khả năng thay thế có khả năng đo các thuộc tính như hoạt động của hệ thống hay người sử dụng đang thử sử dụng phần mềm thay thế cho phần mềm xác định trong môi trường của phần mềm đó.

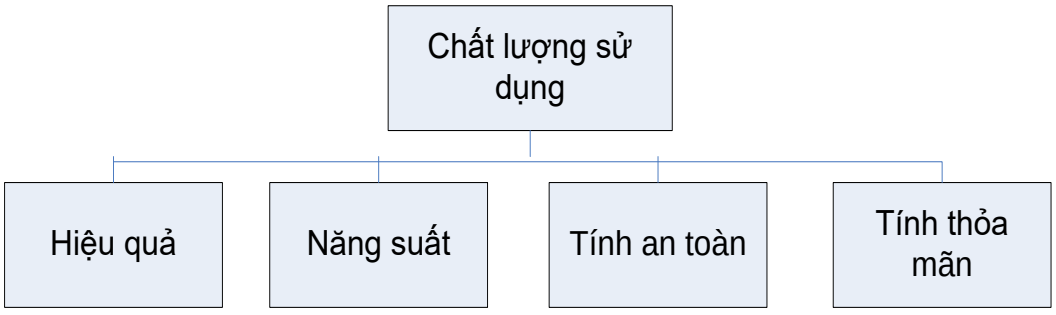
Bảng 22: Bảng các phép đánh giá khả năng thay thế

Tên phép đánh giá	Mục đích của phép đánh giá	Phương pháp áp dụng	Phép đo, công thức và tính toán các thành phần dữ liệu	Chuyển đổi giá trị đo	Đối tượng sử dụng
Khả năng tiếp tục sử dụng dữ liệu	Người sử dụng hay người vận hành có thể dễ dàng tiếp tục sử dụng cùng dữ liệu sau khi thay thế phần mềm này thay cho phần mềm trước không? Liệu việc thay thế hệ thống phần mềm có thành công không?	Quan sát thái độ của người sử dụng hay người bảo trì khi người sử dụng thay thế phần mềm cũ bằng phần mềm mới.	X = A/B A = Khối lượng dữ liệu sử dụng trong phần mềm khác sẽ thay thế và được đảm bảo rằng chúng có khả năng được tiếp tục sử dụng lại. B = Khối lượng dữ liệu sử dụng trong phần mềm khác sẽ được thay thế và dự định tiếp tục được sử dụng lại.	Càng lớn càng tốt	Người phát triển Người bảo trì Người vận hành

Tính bao hàm của chức năng	Người sử dụng hay người vận hành có thể dễ dàng tiếp tục sử dụng chức năng tương tự sau khi thay thế phần mềm không? Liệu việc thay thế phần mềm có thành công không?	Quan sát thái độ của người sử dụng hay người bảo trì khi người sử dụng thay thế phần mềm cũ bằng phần mềm mới.	X = A/B A = Số chức năng tạo ra kết quả tương tự chức năng cũ và không đòi hỏi bất kỳ sự thay đổi nào. B = Số chức năng được kiểm tra tương tự như các chức năng được cung cấp bởi các phần mềm khác bị thay thế	Càng lớn càng tốt	Người phát triển Người bảo trì Người vận hành
Tính nhất quán của chức năng hỗ trợ người sử dụng	Mức độ nhất quán của các thành phần mới với giao diện người sử dụng hiện có như thế nào?	Quan sát thái độ của người sử dụng và tham khảo ý kiến	X = 1 - A/A2 A = Số lượng chức năng mới mà người sử dụng thấy mức độ nhất quán không thỏa đáng với mong muốn B = Số lượng chức năng mới	Càng lớn càng tốt	Người sử dụng Người thiết kế giao diện người sử dụng Người vận hành Người phát triển Người kiểm tra SQA

5. MÔ HÌNH CHẤT LƯỢNG SỬ DỤNG

Chất lượng sử dụng là cách nhìn của người dùng về chất lượng của sản phẩm phần mềm khi nó được sử dụng trong một môi trường và hoàn cảnh cụ thể. Người sử dụng chỉ đánh giá các tiêu chí của phần mềm mà họ dùng tới.



Hình 8: Mô hình chất lượng sử dụng

Trong đó:

- Tính hiệu quả: khả năng của phần mềm cho phép người dùng đạt được mục đích một cách chính xác và hoàn toàn, trong điều kiện làm việc cụ thể.

- Tính năng suất: khả năng của phần mềm cho phép người dùng sử dụng lượng tài nguyên hợp lý tương đối để thu được hiệu quả công việc trong những hoàn cảnh cụ thể.

- Tính an toàn: phần mềm có thể đáp ứng mức độ rủi ro chấp nhận được đối với người sử dụng, phần mềm, thuộc tính, hoặc môi trường trong điều kiện cụ thể.

- Tính thoả mãn: phần mềm có khả năng làm thoả mãn người sử dụng trong từng điều kiện cụ thể.

CHƯƠNG 3: ĐỘ ĐO PHẦN MỀM

1. KHÁI NIỆM

Số đo (Measure): cung cấp một phạm vi, số lượng, kích thước, khả năng của một thuộc tính của một sản phẩm hoặc qui trình phần mềm.

Đo (Measurement): là một hành động xác định số đo.

Độ đo (Metric): là các chỉ số đặc trưng của một đối tượng cho một khía cạnh nào đó.

Độ đo chất lượng phần mềm (Software Quality Metric): Độ đo xác định một số đo định lượng về mức độ mà một phần tử có được một thuộc tính chất lượng đã cho. Một đo lường định lượng. Có thể hiểu một cách khác là một hàm mà đầu vào là dữ liệu về phần mềm và đầu ra là một giá trị số phản ánh mức độ mà phần mềm có được một thuộc tính chất lượng nào đó.

2. TẦM QUAN TRỌNG CỦA ĐỘ ĐO PHẦN MỀM

Độ đo phần mềm là cách tiếp cận định lượng để có cơ sở phân tích, đánh giá một cách khách quan về một vấn đề hay về một đối tượng nào đó.

Khi ghi ngờ, đặt giả thuyết, muốn tìm hiểu, phải thực hiện đo lường để xác định kết quả, thực hiện phân tích kết quả để kết luậ, dự đoán về khía cạnh nghi ngờ.

Tuy nhiên, mỗi số đo không phản ảnh hết mọi khía cạnh của đối tượng mà phải cần phối hợp nhiều độ đo, vận dụng thêm các tiếp cận định tính.

Các độ đo phần mềm dùng để tính toán, ước lượng được các đại lượng liên quan đến các đối tượng, các hoạt động thuộc về tiến trình sản xuất phần mềm. Mục đích đo phần mềm bao gồm:

- Ước lượng giá gia công, phỏng đoán kích thước.

- Đánh giá chất lượng phần mềm.

- Đánh giá chất lượng quy trình sản xuất.

Các mục đích trên nhằm cải tiến chất lượng phần mềm, tiến trình sản xuất phần mềm.

Ví dụ 1: Độ đo sự vừa ý của khách hàng khi sử dụng phần mềm được tính theo công thức như sau:

$$\frac{\text{Số khách hàng vừa ý phần mềm}}{\text{Tổng số khách hàng sử dụng phần mềm}}$$

Độ đo trên phụ thuộc vào quan niệm khách hàng, cụ thể:

- Giá trị đo lớn: phần mềm “chất lượng” tốt.

- Giá trị đo nhỏ: phần mềm“chất lượng” không tốt.

Tuy nhiên, có thể không phản ảnh được “chất lượng bản chất” của phần mềm đang khảo sát do có thể tồn tại ý kiến chủ quan như: ý kiến khảo sát hình thức, nể nang…

Ví dụ 2: Ta có độ đo phần trăm như thông tin sau:

Dạng lỗi	Đề án A	Đề án B	Đề án C
Khảo sát yêu cầu khách hàng	15.0 %	41.0 %	20.3 %
Thiết kế	25.0 %	21.8 %	22.7 %
Mã hóa chương trình	50.0 %	28.6 %	36.7 %
Các lỗi khác	10.0 %	8.6 %	20.3 %
Tổng phần trăm	100.0 %	100.0 %	100.0 %
Tổng số lỗi (tổng số trường hợp)	**200 lỗi**	**105 lỗi**	**128 lỗi**

Tỷ lệ phần trăm lỗi giữa các giai đoạn giúp chúng ta xác định được khâu nào dễ xãy ra lỗi để tìm giải pháp khắc phục. Tuy nhiên, chỉ áp dụng tốt đối với các dự án giống nhau, các dự án khác nhau áp dụng phương pháp khắc phục theo thống kê trên sẽ không có nhiều hiệu quả.

3. CÁC ĐỘ ĐO PHẦN MỀM

Độ đo trực tiếp là các độ đo có thể tính đếm trực tiếp không thông qua các độ đo khác như: Độ đo trực tiếp cho một sản phẩm phần mềm - số dòng lệnh (LOC – Lines Of Code), kích cỡ bộ nhớ, số lỗi,…

Độ đo gián tiếp là độ đo tính qua các độ đo khác như: tỉ lệ lỗi = số lỗi/dòng mã nguồn, FP... Độ đo gián tiếp bao gồm: độ phức tạp, tính hiệu quả, độ tin cậy, tính bảo trì,…

3.1. Độ đo PUM

PUM (Problems per User Month) là độ đo trực tiếp được thực hiện bằng phương pháp tính theo sự cố phản hồi của khách hàng. Công thức tính PUM như Hình 9.

$$PUM = \frac{\text{Tổng sự cố do khách hàng báo cáo}}{\text{Số bản cài đặt x Số tháng khai thác}}$$

Hình 9: Công thức tính PUMP

PUM có mối quan hệ chặt chẽ với chất lượng phần mềm, cụ thể là tỷ lệ nghịch với chất lượng phần mềm.

Hình 10: Mối liên hệ giữa PUM và chất lượng phần mềm

3.2. Độ đo LOC (Lines Of Code)

3.2.1. Tổng quan về độ đo LOC

LOC hay KLOC (nghìn LOC) là độ đo trực tiếp được tính bằng phương pháp đếm số dòng mã lệnh tạo nên chương trình. Độ đo này chỉ có thể chính xác sau khi dự án đã kết thúc. Bên cạnh đó, LOC sau khi kết thúc dự án sẽ được dùng để ước lượng các dự án tương tự sau này.

LOC phụ thuộc vào môi trường lập trình nên khó so sánh giữa các dự án nếu chúng phát triển trên các môi trường lập trình khác nhau. Bên cạnh đó, chưa có sự thống nhất trong quy ước đếm LOC. Trước đây, không tính đến các dòng khai báo dữ liệu, chú thích... nhưng gần đây tính cả dòng khai báo, chú thích vì các dòng này cũng thường xuyên gây lỗi như các dòng lệnh thông thường.

Ví dụ: Ta có chương trình như sau.

1	int sort(int x[], int n)
2	{
3	int i,j, save, im1;
4	/* this function sorts array x
5	if (n<2) return 1;
6	for (i=2;i<=n;i++)
7	{
8	im1= i-1;
9	for(j=1;j<=im;j++)
10	If (x[i] < x[j])
11	{
12	Save = x[i];
13	x[i] = x[j];
14	x[j] = Save;
15	}
16	}
17	Return 0;
18	}

Câu hỏi đặt ra là LOC của chương trình là bao nhiêu? 17, 18 hay 13?

Theo định nghĩa của CONTE như sau: “Dòng mã là bất kỳ dòng nào (tiêu đề, khai báo, lệnh khả thi và không khả thi) trong chương trình, không kể dòng chú thích (comment), bất kể có bao nhiêu dòng mã hay khối mã trên cùng 1 dòng”.

Do đó, LOC của chương trình trên là 17.

Độ đo LOC thường dùng để ước tính thời gian lập trình.

3.2.2. Các độ đo dẫn suất

Căn cứ vào độ đo LOC, có thể tính một số độ đo dẫn suất bao gồm: Năng suất, chất lượng, giá trị…

$$năng\ suất = \frac{\text{tổng số dòng mã nguồn}}{\text{số người tham gia} * \text{thời gian}}$$

$$\text{chất lượng} = \frac{\text{số lỗi}}{\text{LOC}}$$

$$giá\ trị = \frac{\text{chi phí tổng cộng dự án}}{\text{LOC}}$$

Hình 11: Một số độ đo dẫn suất dựa trên LOC

Các độ đo dẫn suất được sử dụng để đo lường các khía cạnh khác nhau của quá trình phát triển phần mềm và chất lượng của sản phẩm phần mềm. Các độ đo này giúp cung cấp thông tin cụ thể về hiệu suất của quy trình phát triển và giúp quản lý dự án và nhóm phát triển hiểu rõ hơn về tiến trình và chất lượng của dự án.

3.2.3. Ưu điểm của độ đo LOC

Đơn giản.

Cụ thể.

Dễ thực hiện.

3.2.4. Hạn chế của độ đo LOC

Phụ thuộc vào ngôn ngữ, không chỉ ra được tổng thể dự án.

Đếm dòng lệnh tương tự như đếm số gạch để xây nhà cao tầng, chỉ cho biết số gạch cần dùng nhưng không chỉ ra được số phòng, tổng diện tích xây dựng, tổng diện tích đất,…

3.2.5. Ví dụ về độ đo LOC

Ta có một mô tả về một dự án phần mềm xác định được các chức năng chính của phần mềm và số dòng lệnh ước tính như sau:

Function	Estimated LOC
User interface and control facilities (UICF)	2,300
Two-dimensional geometric analysis (2DGA)	5,300
Three-dimensional geometric analysis (3DGA)	6,800
Database management (DBM)	3,350
Computer graphics display facilities (CGDF)	4,950
Peripheral control function (PCF)	2,100
Design analysis modules (DAM)	8,400

Mô tả: Số dòng lệnh trung bình là 620 LOC/(người*1 tháng). Tiền thuê lập trình viên là 8,000 USD/(người*1 tháng)

Yêu cầu: Tính độ lớn của phần mềm theo LOC và các độ đo dẫn suất bao gồm năng suất, giá giá trị và chi phí tổng cộng của dự án.

Giải:

Ta có tổng số LOC của dự án là:

2,300 + 5,300 + 6,800 + 3,350 + 4,950 + 2,100 + 8,400 = 33,200 LOC

Năng suất: 33,200/620 ~ 54 (người*1 tháng)

Giá trị 1 LOC: 8,000/620 ~ 14 USD/LOC

Chi phí tổng cộng của dự án: 33,200 * 14 = 464,800 USD

3.3. Độ đo điểm chức năng (độ đo hướng chức năng)

3.3.1. Tổng quan về độ đo điểm chức năng

Độ đo điểm chức năng (Function Point - FP) là độ đo gián tiếp cho phần mềm và tiến trình phát triển phần mềm. Độ đo hướng chức năng tập trung vào "chức năng" hay "tiện ích" của chương trình do đó đo lường quy mô dự án theo FP không phụ thuộc vào việc sử dụng công nghệ.

Ví dụ: Ta có hai phần mềm kế toán, một được viết bằng assembler, một được viết bằng C#, nhưng thực thi những chức năng như nhau. Người dùng chỉ quan tâm đến việc mua phần mềm kế toán, mà không cần quan tâm đến nó chứa bao nhiêu dòng lệnh, sử dụng bằng ngôn ngữ nào.

Qua ví dụ trên, nếu chúng ta tính theo độ đo LOC thì giá trị của 2 phần mềm sẽ khác nhau. Tuy nhiên, nếu đo theo FP thì chúng giống nhau.

Độ đo FP được tính theo 5 yếu tố chính và 14 yếu tố phụ, cụ thể như sau:

- Các yếu tố chính bao gồm:

+ Số input theo người dùng (user input): số các thành phần dữ liệu đưa vào.

+ Số output theo người dùng (user output): các báo cáo, các màn hình, các thông báo lỗi,…

+ Số truy vấn trực tuyến (user online queries) của người dùng: số input trong các truy vấn online.

+ Số tập tin dữ liệu (Logical file): bảng CSDL, các file độc lập.

+ Số các giao tiếp với các hệ thống thông tin khác (external interfaces).

Mỗi yếu tố được chia ra làm 3 loại: Simple, Average, Complex. Các bước tính toán và trọng số các yếu tố được tính theo Hình 12.

Software System Component	Complexity level — 2.evaluate each component									Total CFP
	Simple			Average			Complex			
1.identify functional components	Count	Weight factor	Points	Count	Weight factor	Points	Count	Weight factor	Points	
	A	B	C= A×B	D	E	F= D×E	G	H	I= G×H	J=C+F+I
User inputs		3			4			6		4.CFP is the summed weighted values
User outputs		4			5			7		
User online queries		3			4			6		
Logical files		7			10			15		
External interfaces		5			7			10		
Total CFP	3.apply weighting factors									

Hình 12: Bảng tính giá trị của 5 yếu tố chính

- Các yếu tố phụ bao gồm:

+ Hệ thống đòi hỏi backup và phục hồi an toàn?

+ Đòi hỏi trao đổi dữ liệu truyền thống?

+ Có các chức năng xử lý phân tán?

+ Hiệu năng là điều quan trọng?

+ Yêu cầu sử dụng môi trường nặng về tiện ích?

+ Hệ thống đòi hỏi nhập dữ liệu online?

+ Khi đòi hỏi dữ liệu online, cần bao nhiêu màn hình nhập dữ liệu?

+ File chính được cập nhật online?

+ Có yêu cầu các thao tác nhập xuất hay các câu truy vấn phức tạp?

+ Xử lý bên trong phức tạp

+ Mã được thiết kế dùng lại

+ Việc chuyển đổi dữ liệu và cài đặt hệ thống có được đưa vào trong thiết kế?

+ Hệ thống có được thiết kế để cài đặt lần cho nhiều tổ chức khác nhau hay không?

+ Hệ thống có được thiết kế để dễ thay đổi và dễ dàng sử dụng?

Trong đó mỗi yếu tố phụ nhận giá trị từ 0 đến 5.

Công thức tính FP như sau:

$$\mathrm{FP} = \mathrm{CFP} * (0{,}65 + 0{,}01 * \mathrm{RCAF})$$

Trong đó:

CFP (Crude Function Point): tổng điểm của 05 yếu tố chính

RCAF (Relative Complexity Adjustment Factor): tổng điểm của 14 yếu tố phụ

Ví dụ: Khảo sát dự án với các đơn vị chức năng như sau:

+ Number of user inputs = 50

+ Number of user outputs = 40

+ Number of user enquiries (online queries) = 35

+ Number of user files (logical files) = 06

+ Number of external interfaces = 04

Mô tả: Giả sử tất cả các thừa số điều chỉnh độ phức tạp và thừa số trọng số đều có giá trị trung bình.

Yêu cầu: Tính FP cho dự án.

Giải:

Do các thừa số trọng số đều trung bình nên ta có:

CFP = 50*4 + 40* 5 + 35*4 + 6*10 + 4*7 = 628

Do các thừa số điều chỉnh độ phức tạp đều trung bình nên ta có:

RCAF = 14 * 3 = 42

Vậy $\mathrm{FP} = \mathrm{CFP} * (0{,}65 + 0{,}01 * \mathrm{RCAF}) = 628 * (0{,}65 + 0{,}01 * 42) = 671{,}96$

3.4. Độ đo phức tạp theo chu kỳ

Độ phức tạp theo chu kỳ là phép đo độ phức tạp của mã do McCabe đề xuất, nó cho phép đo độ phức tạp của một chương trình. Nó chỉ ra rằng một hàm có độ phức tạp chu kỳ lớn hơn 10 là khó duy trì do phụ thuộc quá nhiều vào các nhánh,vòng lặp…Nếu các hàm có giá trị lớn hơn 10 thường được thiết kế lại.

Công thức tính độ phức tạp theo chu kỳ là C = E - N + 2, trong đó E là số cạnh của đồ thị, N là số nút và C là độ phức tạp của McCabe.

Để sử dụng độ đo phức tạp theo chu kỳ chúng ta phải mô hình hóa mã nguồn chương trình thành thành đồ thị dòng điều khiển, sau đó áp dụng công thức trên để tính độ phức tạp của chương trình.

Độ phức tạp theo chu kỳ của McCabe là một thước đo đơn giản và trực quan cho phép đạt được kết quả đáng tin cậy ở cấp hàm, chức năng.

3.5. Độ đo của Halstead

Lý thuyết của Halstead được suy ra từ một giả thuyết nền tảng "Bộ óc con người tuân theo một tập hợp chặt chẽ các quy tắc hơn là nó từng biết…". Khoa học phần mềm dùng một tập các cách đo nguyên thuỷ có thể suy ra sau khi mã được sinh ra hay được ước lượng một khi thiết kế đã hoàn thành.

Halstead đưa ra các khái niệm để thực hiện các phép tính, cụ thể:

- Toán tử duy nhất (n1): số các toán tử phân biệt xuất hiện trong chương trình. Toán tử là các phần tử cú pháp như +, -, <,>

- Toán hạng duy nhất (n2): số các toán hạng phân biệt xuất hiện trong chương trình. Toán hạng bao gồm các biểu thức nghĩa đen, hằng và biến.

- Tổng số toán tử (N1): tổng số lần xuất hiện toán tử.

- Tổng số toán hạng (N2): tổng số lần xuất hiện toán hạng.

- Từ vựng (n): Số lượng toán tử và toán hạng duy nhất trong chương trình. Đây là ước tính kích thước của từ vựng của chương trình.

- Độ dài (N): Độ dài chương trình.

Các công thức trong phép đo của Halstead:

- Công thức tính số từ vựng: n=n1+n2.

- Công thức chiều dài: N = N1 + N2.

- Công thức ước lượng chiều dài N' = n1log2n1 + n2log2n2.

- Công thức tính khối lượng chương trình (độ lớn) V = Nlog2(n1 + n2).

- Cấp độ chương trình (program level) L=V*/V; V* là biểu diễn tối thiểu nhất của giải thuật đang xét. Tuy nhiên, L thường được tính bởi L'=(2/n1)(n2/N2).

- Công sức viết chương trình: E=V/L.

- Thời gian viết chương trình: T=E/18 giây.

Ví dụ: Cho chương trình sau:

```
Procedure sort(var x:array; n:integer);
    Var i,j,save:integer;
        Begin
            for i:=2 to n do
            for j:=1 to i do
                    if x[i]<x[j] then
                    begin
                        save:=x[i];
                        x[i]:=x[j];
                        x[j]:=save
                    end
                End;
```

Yêu cầu: Tính các giá trị theo độ đo Halstead.

- Tính n1 và N1, kết quả như sau:

Operator	**Số lần xuất hiện**
Procedure	1
Sort	1
()	1
Var	2
:	3
Array	1
;	6
Integer	2
,	2
Begin end	2
For do	2
If then	1
:=	5
<	1
[]	6

Ta có n1=15, N1=36.

- Tính n2 và N2, kết quả như sau:

Operand	Số lần xuất hiện
X	7
N	2
I	6
J	5
Save	3
“2”	1
“1”	1

Ta có n2=7, N2=25.

- Áp dụng công thức tính:

+ Size of vocabulary: n= 15 + 7 = 22

+ Program length: N= 36 + 25 = 61

+ Estimated program length N’= $15*\log_2 15 + 7*\log_2 7 = 78.3$

+ Program volume V= $61*\log_2(15 + 7) = 272$

+ Estimated level of abstraction L’= =(2/15)(7/25).= 0.037

+ Programming effort E= 272/ 0.037 = 7351.4

+ Time T= 7351.4/18 = 408 giây

4. CHỈ SỐ CHẤT LƯỢNG CẤU TRÚC

Chỉ số chất lượng về cấu trúc thiết kế - DSQI (Design Structured Quanlity Index – IEEE Standard 982.1-1988) có gái trị chạy từ 0 đến 1.

Cách tính thực hiện như sau, đầu tiên tính các giá trị sau:

+ S1 = tổng số các module (được định nghĩa trong kiến trúc chương trình).

+ S2 = số các module có chức năng tạo ra dữ liệu hay phụ thuộc vào nguồn dữ liệu (không tính các module điều khiển).

+ S3 = số các module có chức năng phụ thuộc trước hết vào xử lý.

+ S4 = số các khoản mục cơ sở dữ liệu.

+ S5 = tổng số các khoản mục cơ sở dữ liệu độc lập.

+ S6 = số các đoạn trong cơ sở dữ liệu (các bản ghi khác nhau hay các sự vật riêng lẻ).

+ S7 = số các module với một dữ liệu vào một dữ liệu ra.

Sau đó tính các giá trị Di như sau:

+ Cấu trúc chương trình D1:Nếu thiết kế kiến trúc được phát triển bằng cách dùng một phương pháp phân biệt (VD: thiết kế hướng luồng dữ liệu, thiết kế hướng sự vật…) thì D1 = 1; ngoài ra D1 = 0

+ Tính độc lập module D2 = 1 – (S2 / S1)

+ Module không phụ thuộc vào xử lý trước D3 = 1 – (S3 / S1)

+ Kích cỡ cơ sở dữ liệu D4 = 1 – (S5 / S4)

+ Việc phân ngăn cơ sở dữ liệu D5= 1 – (S6 / S4)

+ Đặc trưng vào/ra module D6 = 1 – (S7 / S1)

Cuối cúng chúng ta tính DSQI theo công thức:

$$DSQI = \sum_{i=1}^{6} W_i D_i$$

Trong đó Wi là trọng số tương đối về tầm quan trọng của của mỗi giá trị trung gian, tổng Wi =1. Nếu tất cả Di đều có trọng số ngang nhau thì Wi = 0.167

Ý nghĩa của DSQI: dựa vào các giá trị DSQI của quá khứ và so sánh với giá trị DSQI hiện tại. Nếu DSQI hiện tại không bằng DSQI quá khứ thì thiết kế hiện hành không hiệu quả như lần trước và cần phải có thay đổi thiết kế hiện hành.

CHƯƠNG 4: MÔ HÌNH ISO/IEC 14598

1. VÒNG ĐỜI PHẦN MỀM

Vòng đời của một sản phẩm phần mềm theo góc nhìn quản lý chất lượng bắt đầu từ các bài toán thực tiễn và được thể hiện ở qui trình (9 bước) như sau:

a. Từ bài toán thực tiễn, nhu cầu về phần mềm hình thành;

b. Nhu cầu này được thể hiện qua các tài liệu yêu cầu;

c. Nhu cầu sẽ xác định yêu cầu chất lượng ngoài. Thỏa mãn được yêu cầu chất lượng này sẽ thỏa mãn được yêu cầu của người sử dụng;

d. Các yêu cầu chất lượng được thể hiện trong tài liệu đặc tả hệ thống;

e. Yêu cầu chất lượng ngoài là tiền đề cho yêu cầu chất lượng nội bộ;

f. Trong quá trình thiết kế phần mềm, các yêu cầu chất lượng nội bộ được thể hiện trong các đặc tính của phần mềm và chuyển thành chất lượng nội bộ;

g. Ứng với chất lượng nội bộ có các độ đo chất lượng nội bộ phần mềm phải vượt qua;

h. Tới giai đoạn tích hợp chạy thử, chất lượng ngoài sẽ là vấn đề được quan tâm. Phần mềm được gọi là có chất lượng khi tất cả các độ đo được đảm bảo;

i. Trong quá trình vận hành, vẫn sử dụng các độ đo ngoài, chất lượng phần mềm trong quá trình vận hành sẽ tiếp tục được xem xét và cải tiến; 10. Quá trình cải tiến sẽ liên tục diễn ra cho tới khi phần mềm trở nên lạc hậu hoàn toàn, cần được thay thế bằng phần mềm mới.

Với cách tiếp cận như vậy, qui trình đánh giá chất lượng phần mềm được xây dựng qua bốn bước (Chuẩn ISO/IEC 14598).

2. GIỚI THIỆU MÔ HÌNH

Mô hình ISO/IEC 14598 (gọi chung là ISO 14598) xác định quy trình đánh giá chất lượng phần mềm như Hình 13.

Hình 13: Các bước thực hiện trong mô hình ISO 14598

Mô hình được thực hiện qua 4 bước bao gồm:

➢ Bước 1: Thiết lập yêu cầu đánh giá. Bước này được chia thành các bước nhỏ như sau:

- Xác lập mục đích đánh giá: làm rõ mục đích của việc đánh giá phần mềm bằng cách xem xét ai muốn thực hiện loại đánh giá nào và ở giai đoạn nào. Ví dụ, một quản lý dự án trong giai đoạn thiết kế có thể muốn biết mức tuân thủ hướng dẫn thiết kế màn hình theo yêu cầu của khách hàng. Hoặc, một kế hoạch sản phẩm có thể muốn biết sự hài lòng của người dùng sau khi giao hàng để chuẩn bị cho việc cải thiện sản phẩm tiếp theo.

- Xác định loại sản phẩm: Dựa trên mục đích của các bên liên quan, xác định sản phẩm hoặc sản phẩm nào sẽ được đánh giá và ở giai đoạn nào của vòng đời phần mềm. Thông thường, các mục tiêu đánh giá trong giai đoạn thiết kế và triển khai là các thông số kỹ thuật và mã nguồn, trong giai đoạn kiểm thử là kết quả hoạt động của phần mềm thực thi và những giai đoạn vận hành và bảo trì là những hiệu ứng được phần mềm tạo ra đối với người sử dụng.

- Xây dựng mô hình chất lượng: Xác định rõ các đặc điểm và phân loại đặc điểm chất lượng cần thiết cho sản phẩm phần mềm dựa trên các mô hình chất lượng trong ISO/IEC 9126-1... Nhiệm vụ này bao gồm việc lựa chọn những đặc điểm cần xem xét, ưu tiên và bổ sung những đặc điểm còn thiếu. Ngoài ra, cần xem xét khả năng sử dụng để xác định sự cần thiết của việc đánh giá khả năng sử dụng và ưu tiên các đặc điểm liên quan để triển khai và đánh giá chất lượng theo yêu cầu.

➢ Bước 2: Xác lập cơ chế đánh giá. Bước này được chia thành các bước nhỏ như sau:

- Xác định phép đánh giá: Xác định rõ các chỉ số chất lượng sẽ được sử dụng để đánh giá từng đặc điểm trong mỗi giai đoạn. Những chỉ số này nên được lựa chọn từ danh sách các chỉ số được liệt kê trong ISO/IEC 9126-2, -3 và -4, sau khi xem xét lại dựa trên các đặc tính của phần mềm hoặc dự án và được định nghĩa cụ thể cho các mức thực tế. Nếu việc lựa chọn chỉ số từ ISO/IEC 9126-2, -3 và -4 không đủ, cũng có thể định nghĩa và thực hiện thêm vào các chỉ số gốc.

- Thiết lập mức đo chuẩn: Xác định các cấp độ đánh giá cho mỗi chỉ số để quyết định mức độ hài lòng của các giá trị thu được. Điều này là việc làm rõ phạm vi chấp nhận được của các giá trị và các giới hạn không cho phép vượt quá chúng. Giá trị nên là lớn nhất có thể, nhỏ nhất có thể hoặc gần như bằng một giá trị cụ thể, phụ thuộc vào từng chỉ số.

- Thiết lập các tiêu chí đánh giá: Trong mỗi giai đoạn yêu cầu quyết định "Tiến hành - Không tiến hành", ví dụ như quyết định chuyển tiếp giai đoạn hoặc phát hành sản phẩm, làm rõ các tiêu chí đánh giá dựa trên dữ liệu chất lượng thu thập được bằng cách sử dụng các chỉ số khác nhau cũng như dữ liệu về tiến độ hoặc chi phí, nếu cần thiết. Các tiêu chí có thể được thiết lập dựa trên điểm tổng thể thu được thông qua trung bình có trọng số hoặc sử dụng bảng quyết định.

Kết quả tại bước này đưa ra được các tiêu chí đánh giá, mức độ chấp nhận các thuộc tính chất lượng của các đối tượng.

➢ Bước 3: Thiết kế: lập kế hoạch đánh giá.

Lập kế hoạch cho việc đánh giá chất lượng, bao gồm thời điểm và loại dữ liệu nào được thu thập bởi ai và cũng như thời điểm dữ liệu được tổng hợp, đánh giá và đánh bình giá theo các chỉ số và tiêu chí đã được định nghĩa ở trên.

➢ Bước 4: Thực hiện đánh giá. Bước này được chia thành các bước nhỏ như sau:

- Thực hiện đo: Thu thập dữ liệu theo kế hoạch và các chỉ số đã được định nghĩa ở trên và tính toán giá trị đo lường của các chỉ số.

- So sánh với tiêu chí đặt ra: Đánh giá giá trị đo lường tính toán của mỗi chỉ số so với tiêu chí.

- Đánh giá kết quả thu được: Đánh giá chuỗi giá trị đo lường dựa trên tiêu chí đánh giá và thực hiện các quyết định quản lý.

CHƯƠNG 5: KIỂM SOÁT CHẤT LƯỢNG PHẦN MỀM VỚI CÔNG CỤ QUẢN LÝ KIỂM THỬ TESTLINK

1. GIỚI THIỆU VỀ TESTLINK

TestLink là công cụ quản lý được sử dụng rộng rãi dựa trên mã nguồn mở. Nó kết hợp đồng thời cả hai Requirements specification (yêu cầu đặc tả) và Test specification (kiểm tra đặc tả). Người dùng có thể tạo một Test project và tài liệu Test Case sử dụng công cụ này. TestLink hỗ trợ tạo tài khoản cho nhiều người dùng và gán (assign) những quyền người dùng khác nhau nhằm mục đích hỗ trợ người dùng quản lý Test Case. Với công cụ này thì người kiểm thử có thể sử dụng để xuất ra tập tin Test report và tài liệu Test plan nhanh chóng. Nó hỗ trợ xuất ra nhiều định dạng tập tin Test report như: Word, Excel, HTML formats…

TestLink cũng tương thích với các công cụ quản lý lỗi như Mantis, Bugzilla…để tạo thành một chu trình khép kín giữa quản lý kiểm thử và quản lý lỗi.

2. CÀI ĐẶT TESTLINK

2.1. Cài đặt XAMPP

XAMPP là một phần mềm cho phép giả lập môi trường Server Hosting ngay trên máy tính cá nhân, cho phép chạy demo Website mà không cần phải mua Hosting. Chính vì vậy, XAMPP hay được phục vụ cho hoạt động học tập giảng dạy thực hành và phát triển Website. XAMPP được dùng để xây dựng và phát triển Website theo ngôn ngữ PHP. Ngoài ra, XAMPP còn được sử dụng để phát triển, nghiên cứu Website thông qua localhost của máy tính cá nhân, biến máy tính cá nhân thành máy chủ, dùng chính ổ cứng của máy tính để làm nơi lưu trữ cho máy chủ trang web.

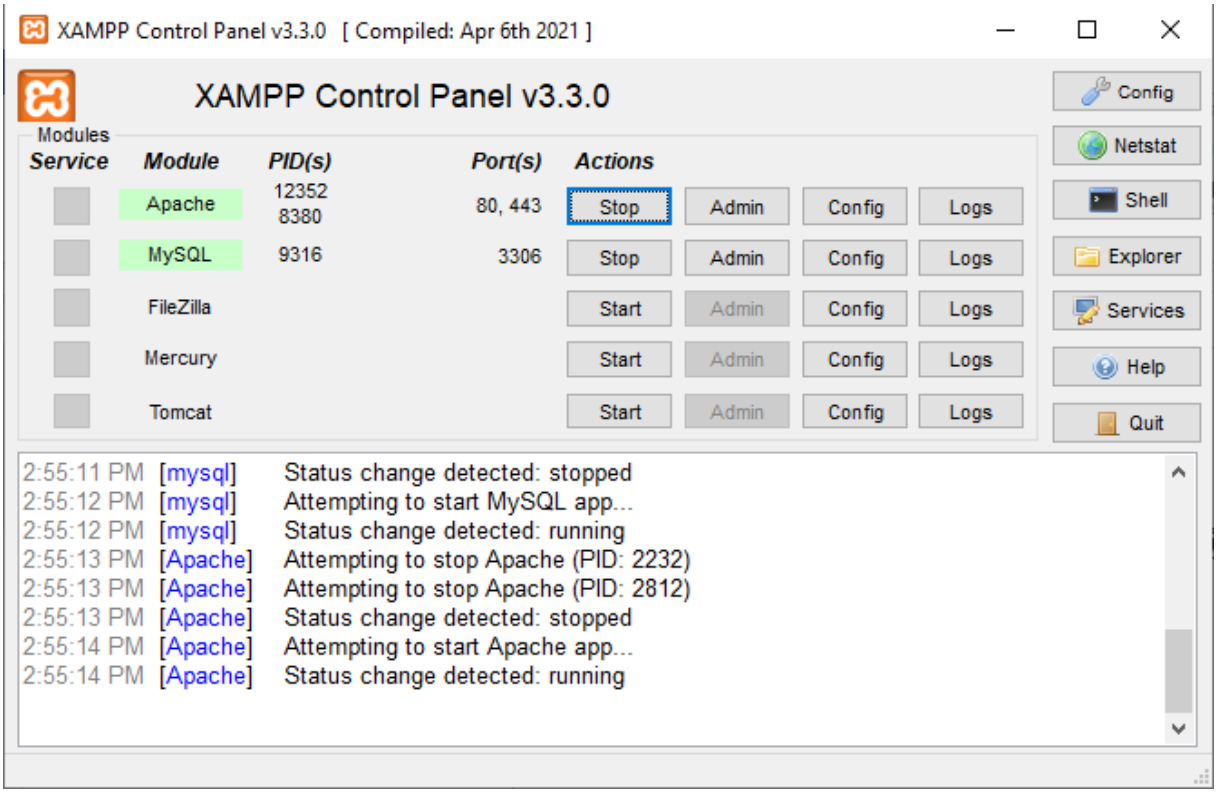
TestLink là công cụ miễn phí mã nguồn mở chạy trên nền web nhưng chỉ hỗ trợ php 5.0. Do đó, khi cài đặt TestLink phải lưu ý nội dung này. Nhóm tác giả đề xuất sử dụng phiên bản xampp-win32-5.6.36-0-VC11.

Việc cài đặt XAMPP khá dễ thực hiện, chỉ cần kích hoạt tập tin cài đặt và chọn các yêu cầu để hoàn tất cài đặt.

2.2. Cài đặt TestLink

Nhóm tác giả đề xuất sử dụng TestLink phiên bản 1.9.16. Sau khi tải và giải nén tập tin cài đặt TestLink, thực hiện tuần tự các bước như sau:

- Bước 1: Chép thư mục TestLink-1.9.16 vào htdocs của XAMPP.
- Bước 2: Kích hoạt XAMPP, chạy dịch vụ Apache và MySQL như Hình 14.

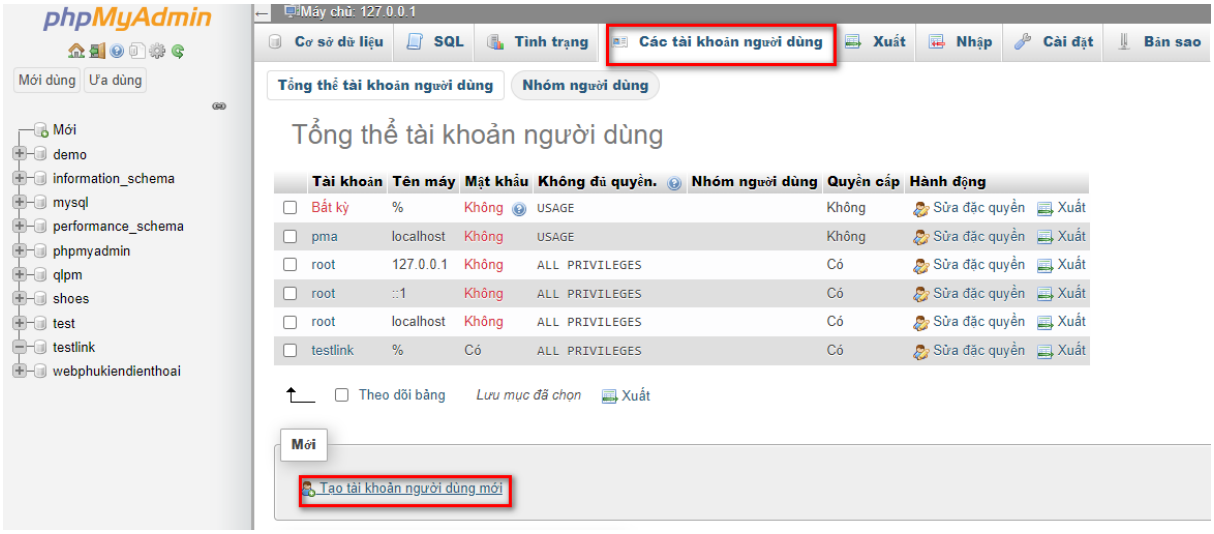


Hình 14: Chạy Apache và MySQL

- Bước 3: chạy chức năng admin của MySQL vào tạo cơ sở dữ liệu tên TestLink như Hình 15.

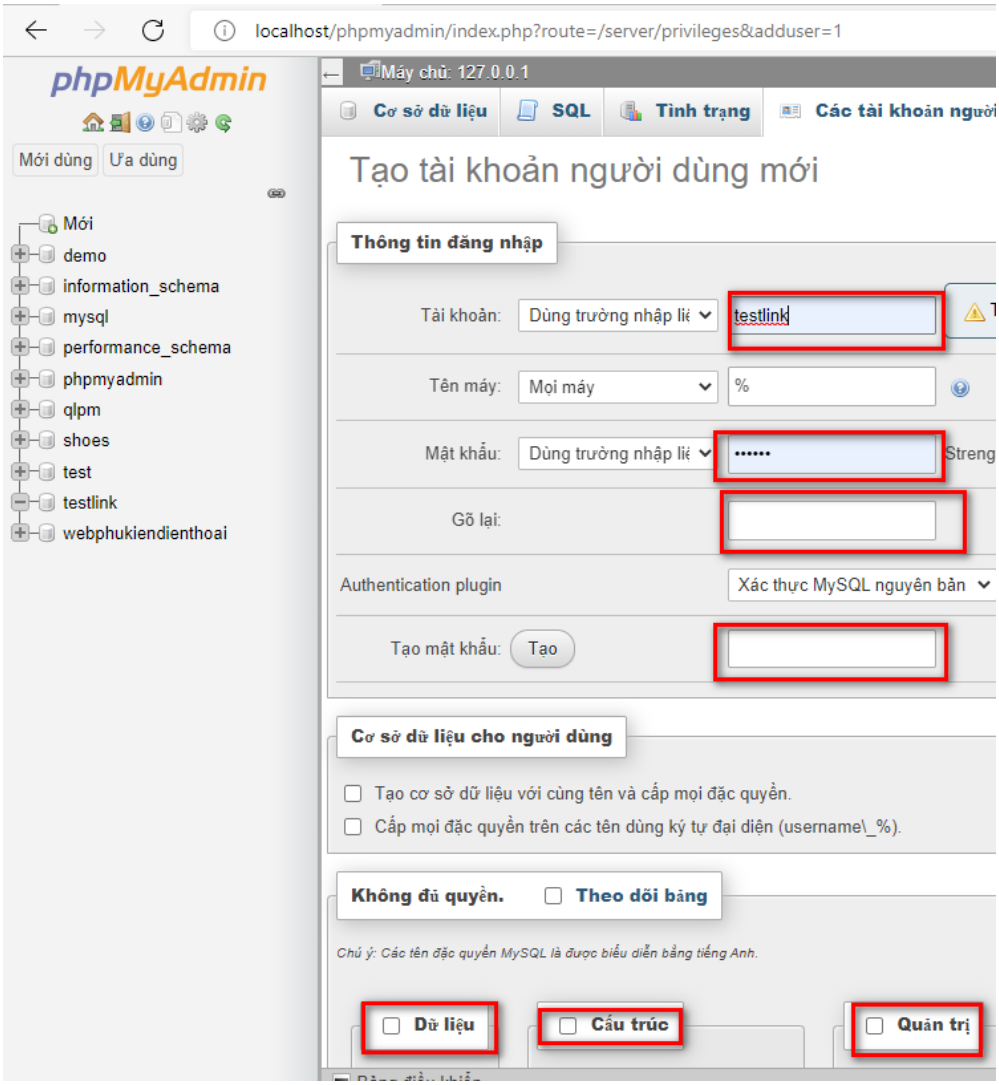
Hình 15: Tạo CSDL TestLink

- Bước 4: Tạo người dùng mới như Hình 16.



Hình 16: Tạo người dùng mới

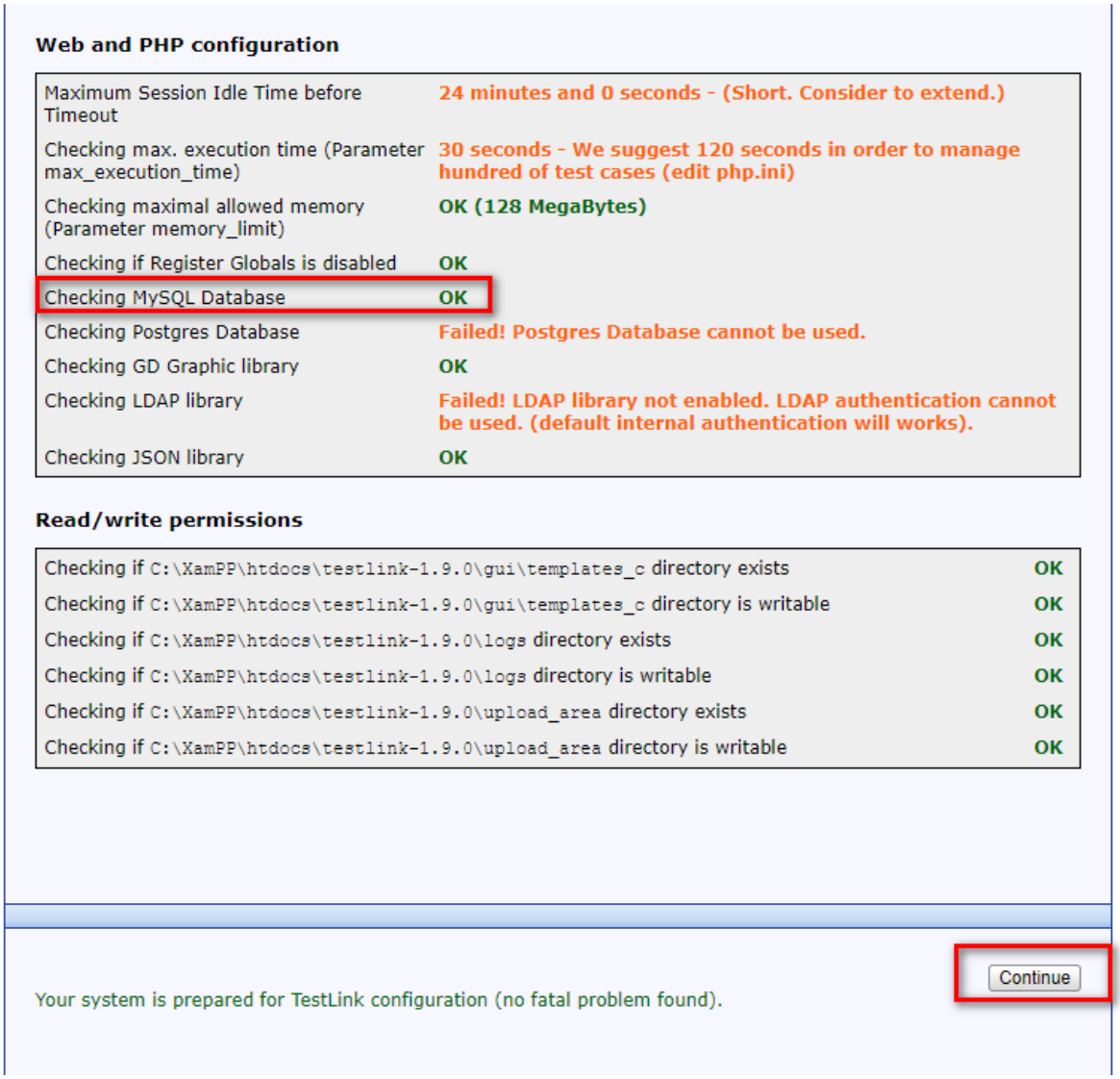
- Bước 5: Tạo tên, mật khẩu và gán tất cả quyền như Hình 17.



Hình 17: Tạo user và Grant

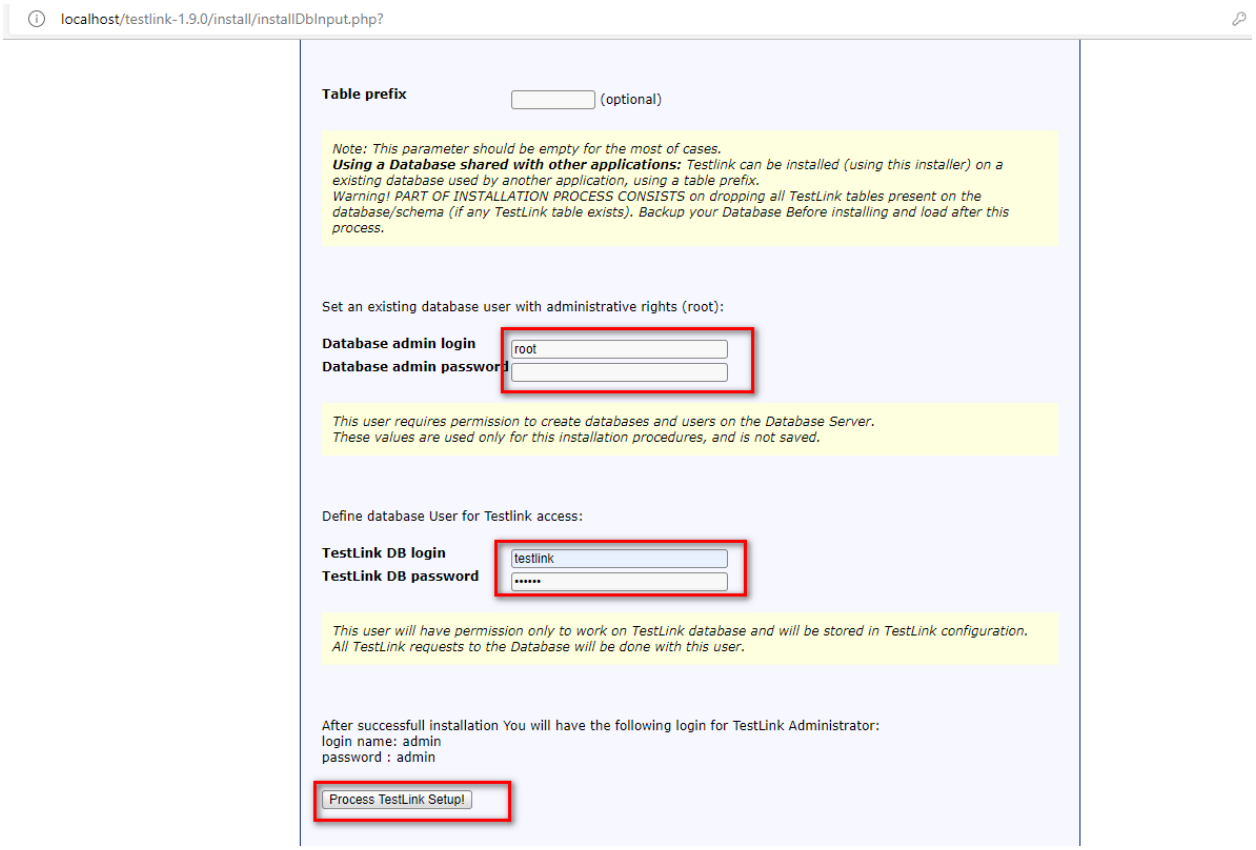
- Bước 6: gõ đường dẫn http://localhost/TestLink-1.9.16/install/index.php và chọn New Installation.

- Bước 7: Kiểm tra Checking MySQL Database và ấn nút Continue.

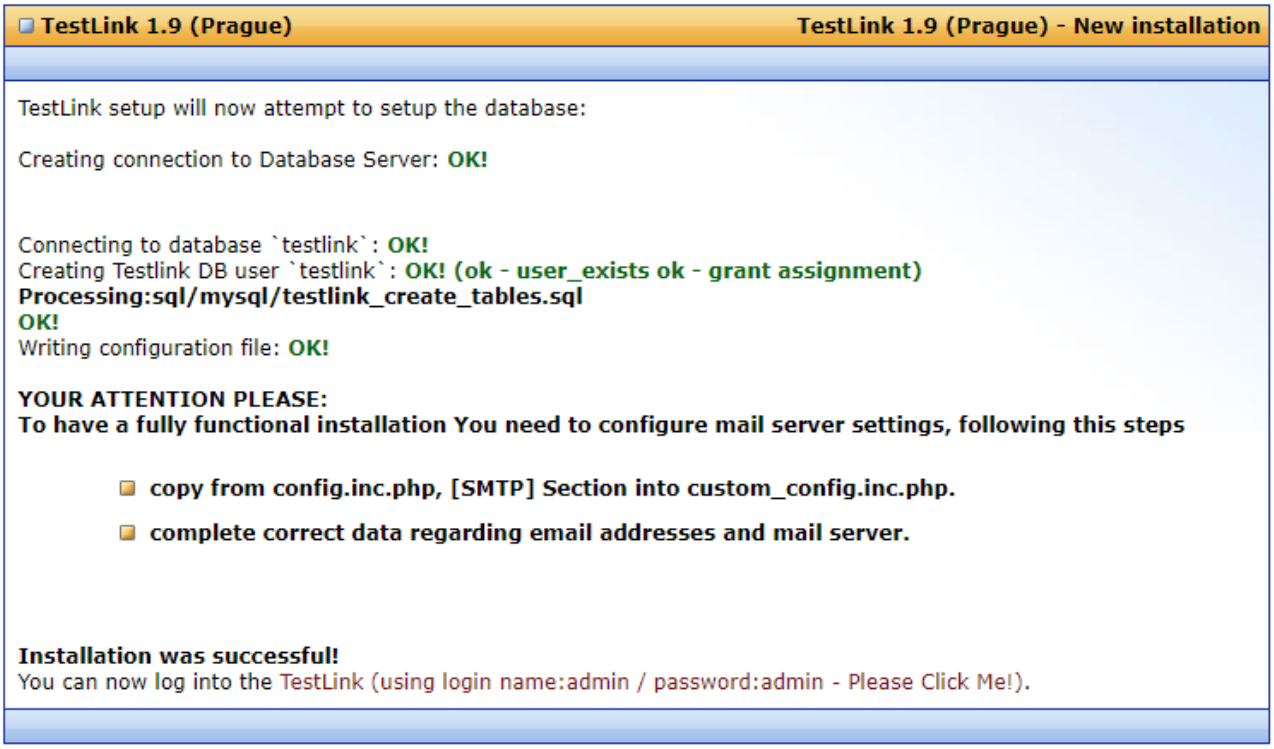


Hình 18: Kiểm tra connect MySQL

- Bước 8: Hoàn tất setup và cấu hình các thông tin như Hình 19. Trong đó TestLink DB là CSDL đã tạo trong MySQL. Cài đặt thành công như Hình 20.



Hình 19: Process TestLink Setup



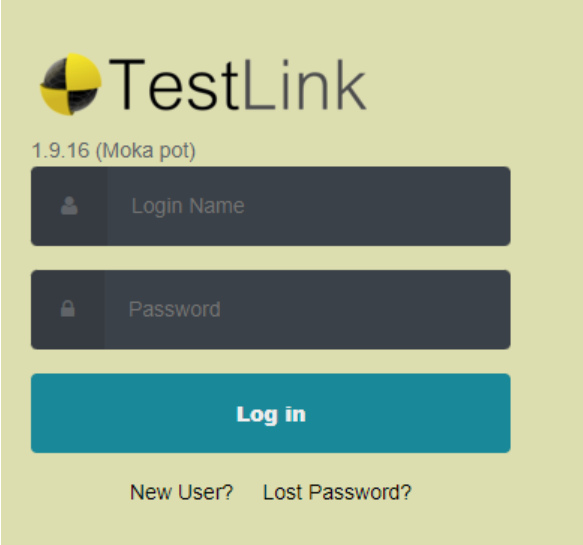
Hình 20: Setup successful

3. HƯỚNG DẪN SỬ DỤNG TESTLINK

Trong hướng dẫn này, sẽ đề cập đến các khía cạnh khác nhau của TestLink như cách có thể sử dụng TestLink làm Test Management. Nó giải thích từng bước cách quản lý Test Plan cho dự án, cách tạo người dùng và gán cho họ vai trò phù hợp hoặc thậm chí cách Import hoặc Export các Test case cho dự án. Các tính năng hữu ích khác như tạo báo cáo, xác định yêu cầu, v.v. cũng được thể hiện tốt trong hướng dẫn này. Các bước sử dụng được nhóm tác giả tổ chức cụ thể, chi tiết để người đọc có thể dễ dàng tiếp cận.

3.1. Đăng nhập TestLink

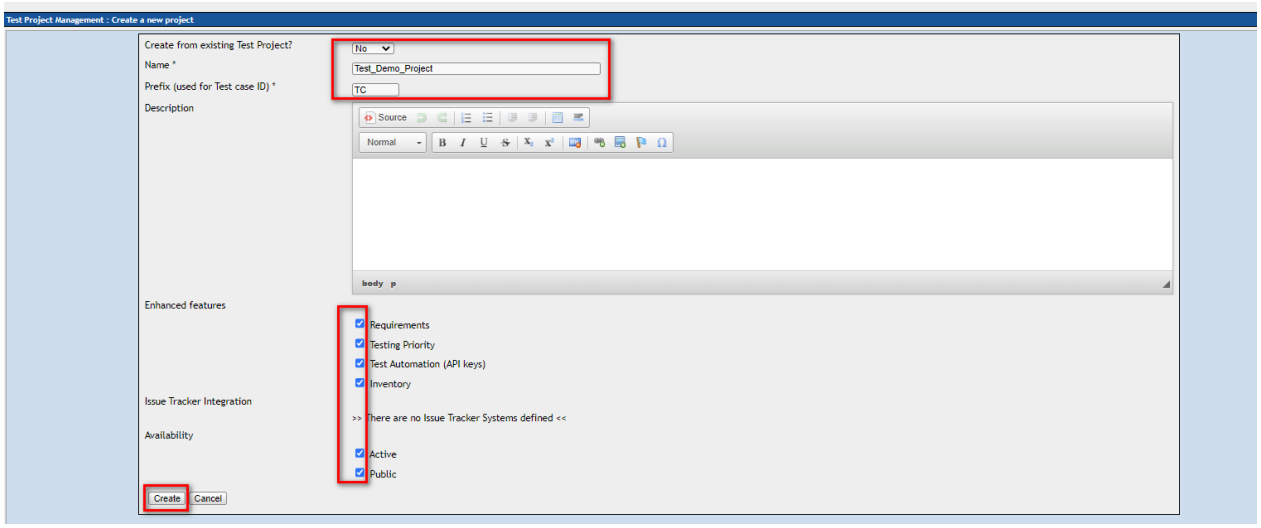
- Bước 1: Đăng nhập thông qua địa chỉ: *http://localhost/TestLink-1.9.16*
- Bước 2: Nhập tài khoản: admin, mật khẩu: admin. Chọn Login.



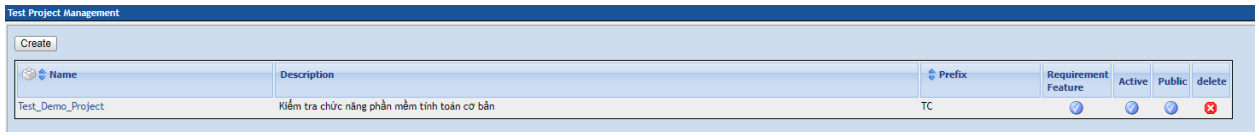
Hình 21: Đăng nhập TestLink

3.2. Tạo Test Project

Khi đăng nhập vào TestLink, nếu chưa có Project nào được tạo thì người dùng bắt buộc phải tạo Project mới trước khi sử dụng. Người dùng phải nhập tất cả các trường bắt buộc trong cửa sổ vừa mở ra như: Category cho project, Tên của project, Prefix, Description, vv... và ấn nút **Create** như Hình 22. Kết quả tạo thành công như Hình 23.



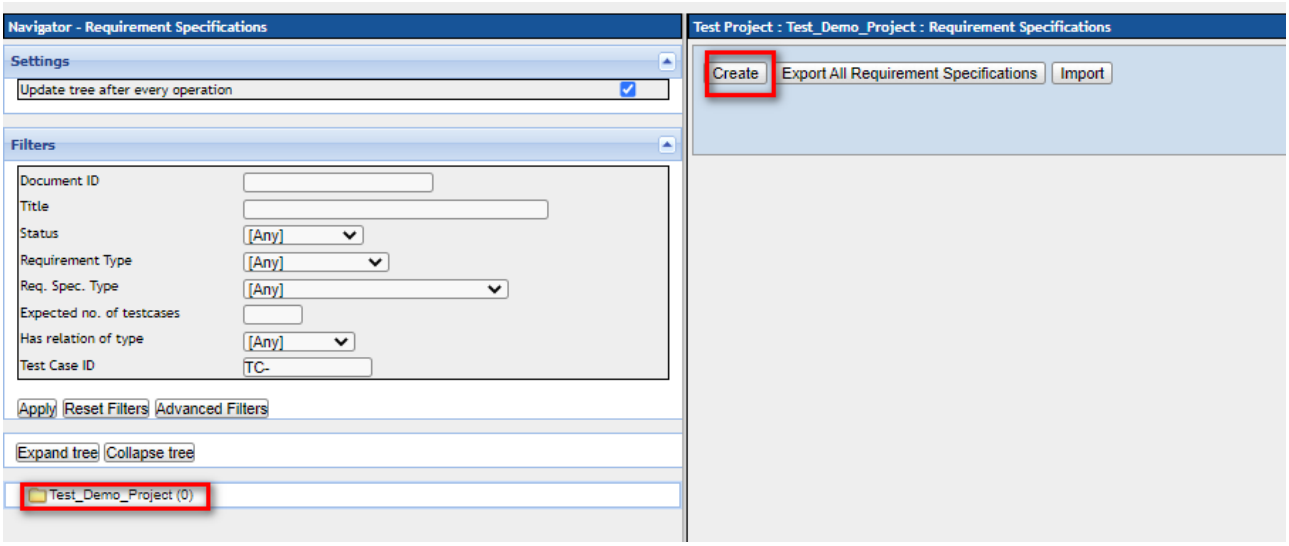
Hình 22: Tạo project



Hình 23: Thông tin Project vừa tạo

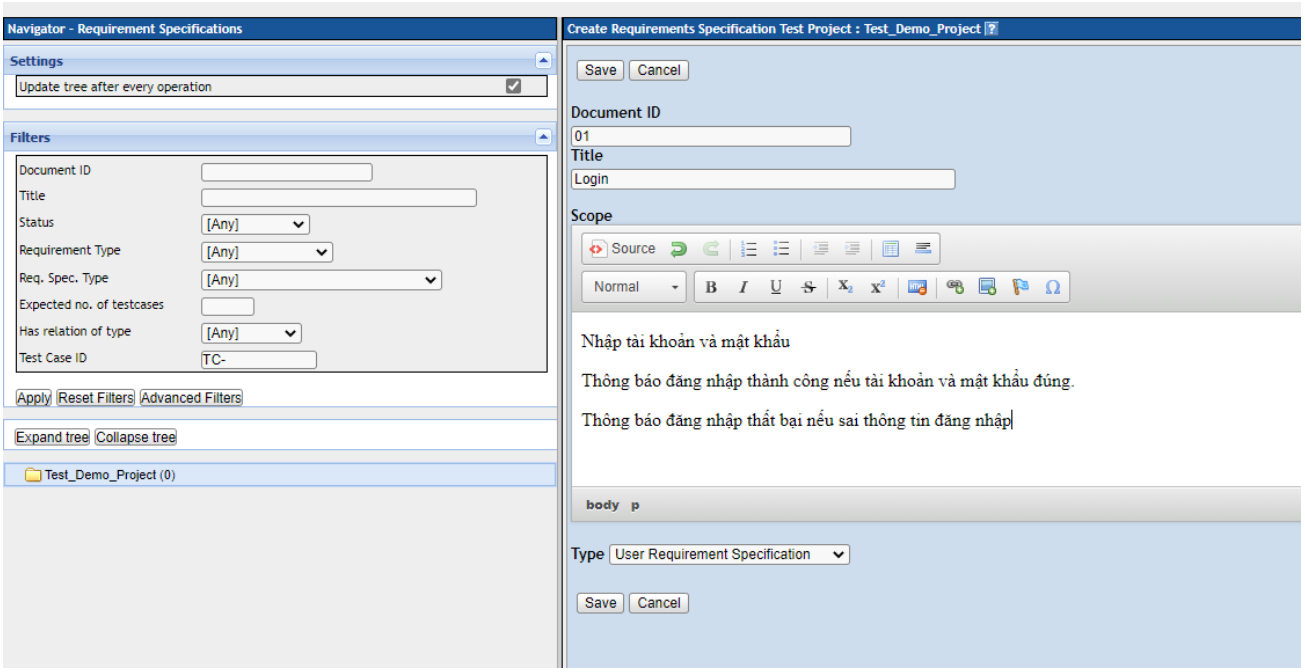
3.3. Viết yêu cầu

Yêu cầu là cơ sở thực hiện kiểm thử. Để viết yêu cầu phần mềm chúng ta chọn thẻ **Requirement Specification** trên menu. Chọn thư mục **Project** và ấn **Create**.

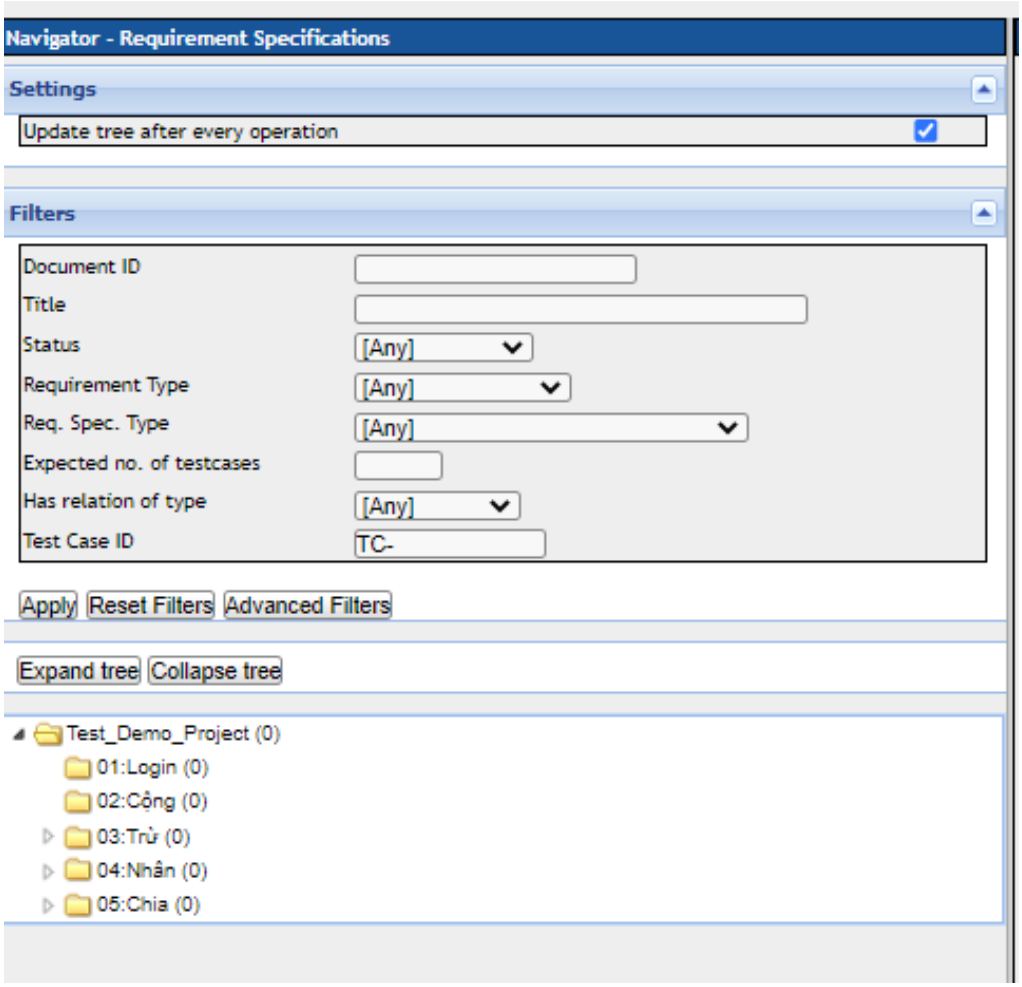


Hình 24: Tạo Requirements

Nhập tất cả thông tin yêu cầu như hình 5. Ấn nút **Save**. Danh sách yêu cầu hiển thị như Hình 25.

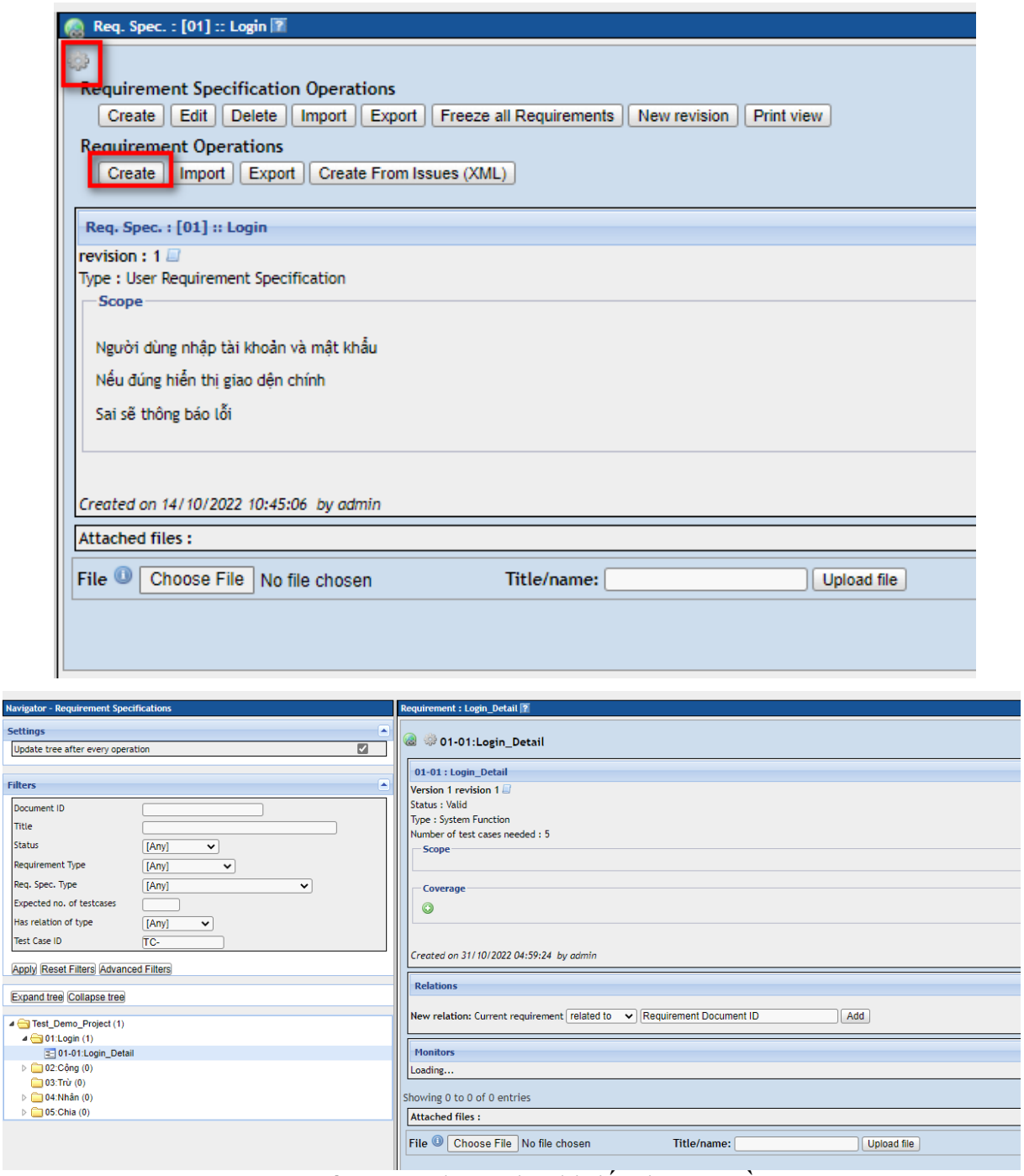


Hình 25: Chi tiết thông tin yêu cầu người dùng



Hình 26: Danh mục Requirement

Đối với từng yêu cầu người dùng, chúng ta phải chi tiết hóa các yêu cầu bằng cách chọn yêu cầu -> **Requirement Operations** -> **Create**. Điền các thông tin như Hình 27.

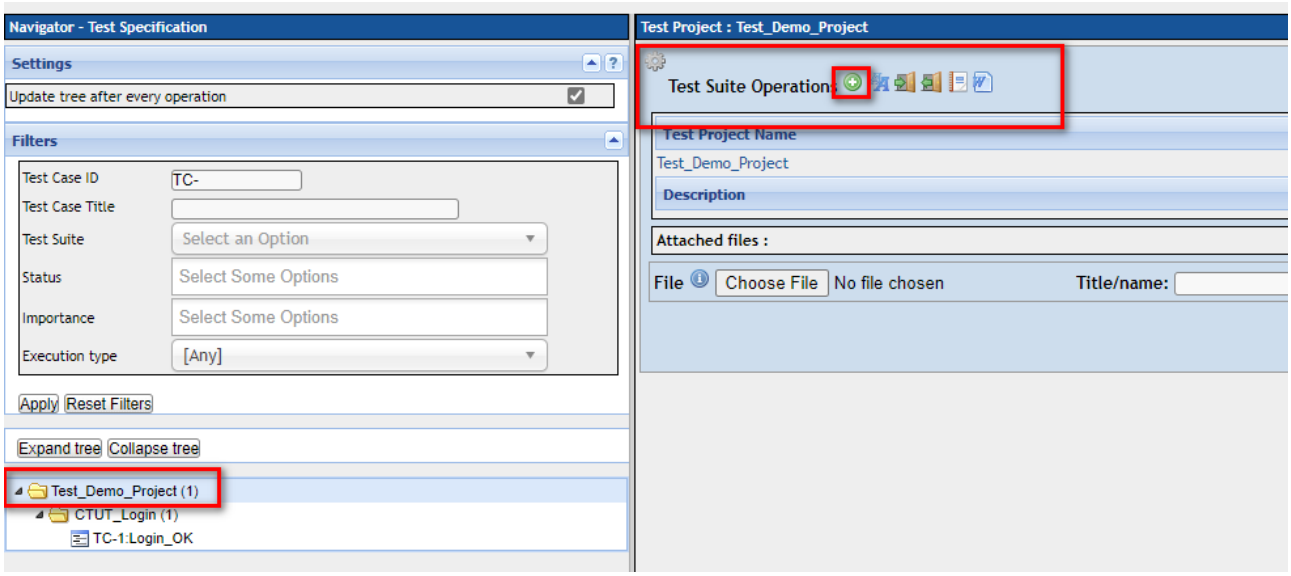


Hình 27: Thông tin chi tiết cho yêu cầu

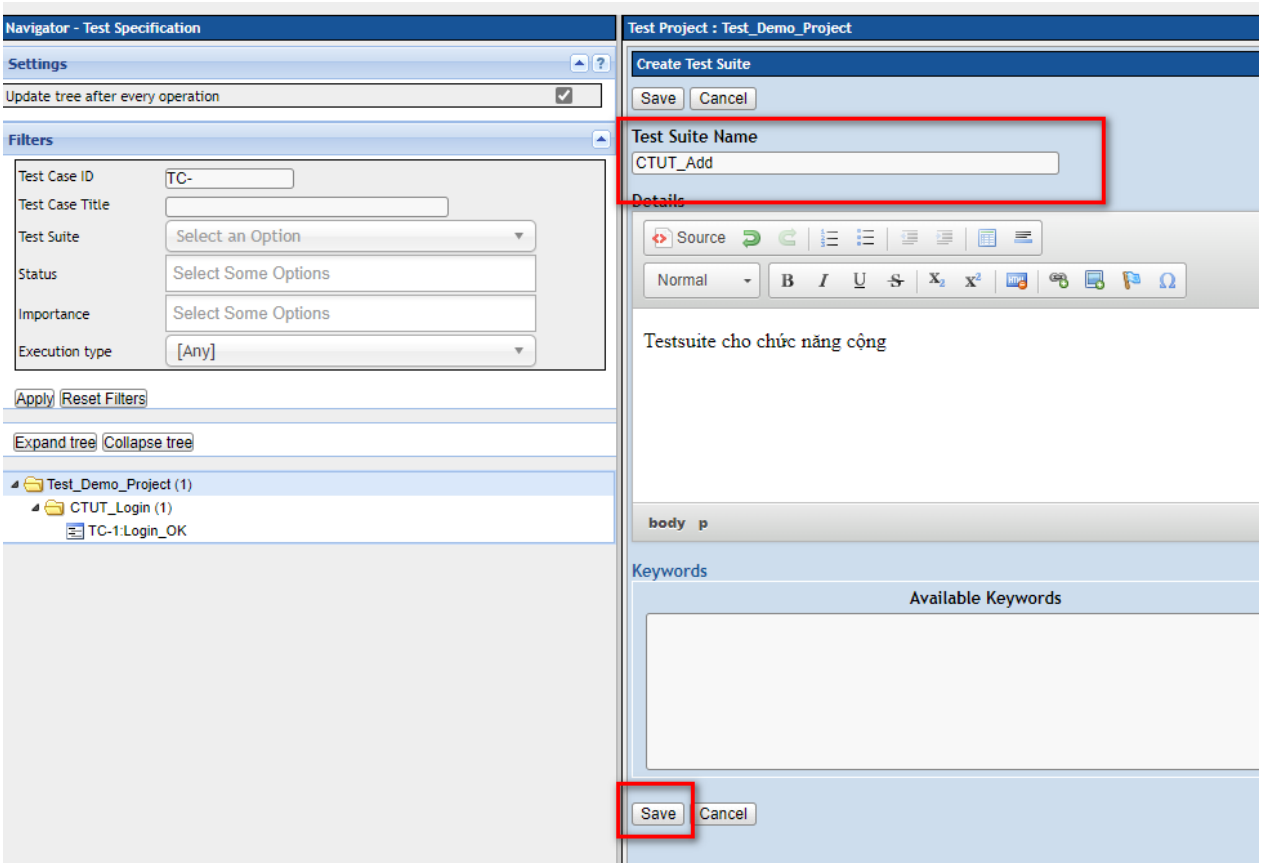
3.4. Tạo Test Suite

Test Suite còn được gọi là bộ kiểm thử (chứa các Test Case) cho 1 chức năng nào đó của phần mềm nhằm nâng cao hiệu quả quản lý và thực thi kiểm thử. Để tạo Test Suite, truy cập vào chức năng Test specification, chọn Project và chọn Test Suite Operation như Hình

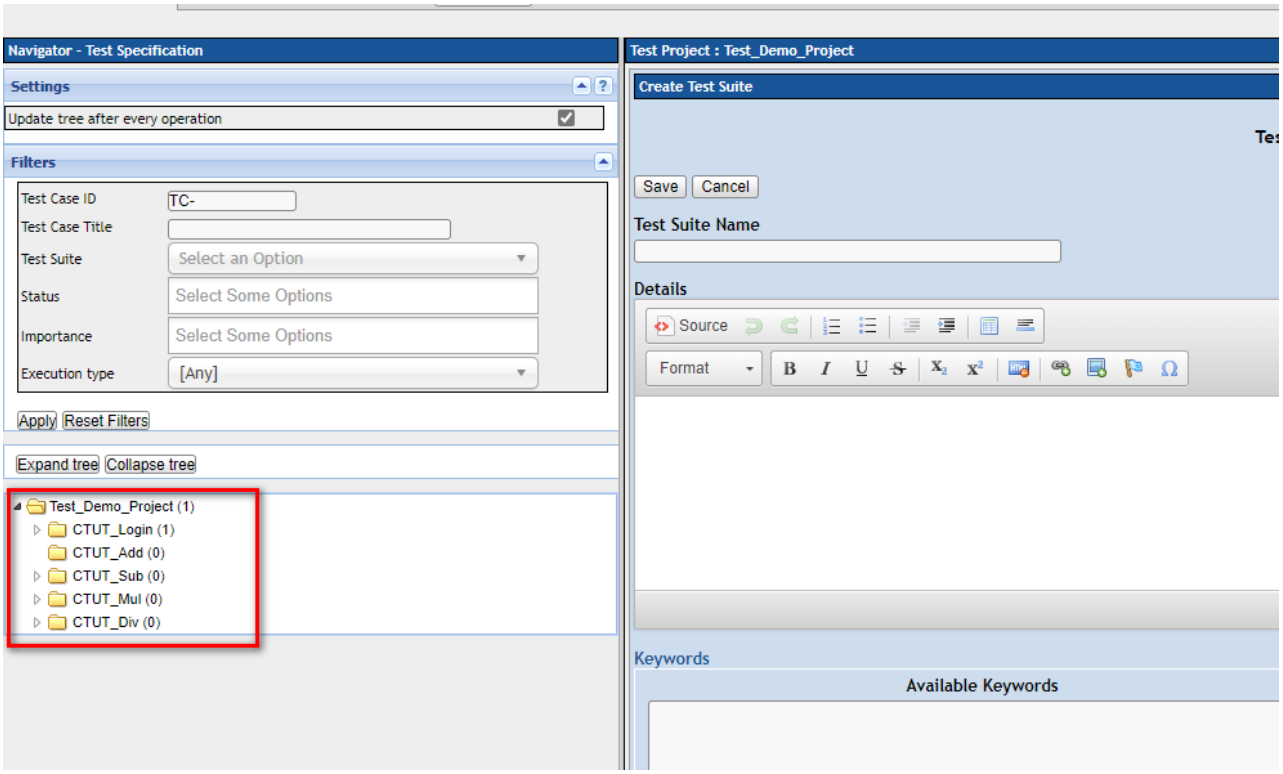
28, điền các thông tin cho Test Suite như Hình 29 và kết quả sau khi tạo Test Suite như Hình 30.



Hình 28: Tạo mới Test Suite



Hình 29: Submit tạo mới Test Suite

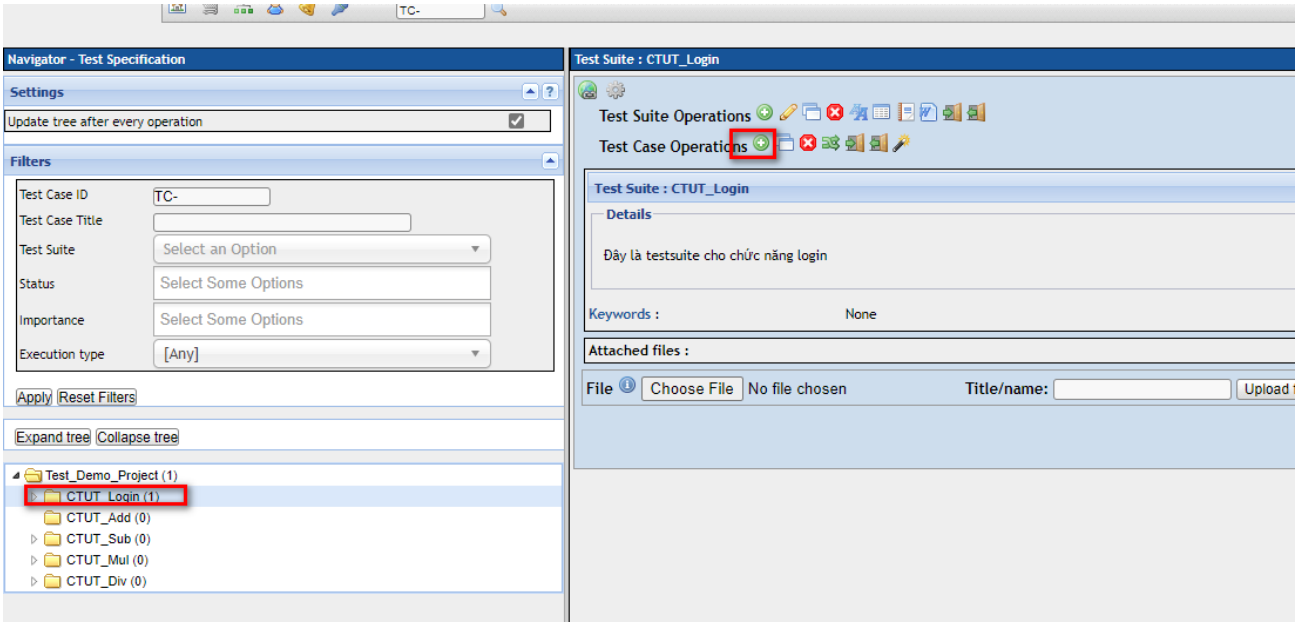


Hình 30: Danh sách Test Suite

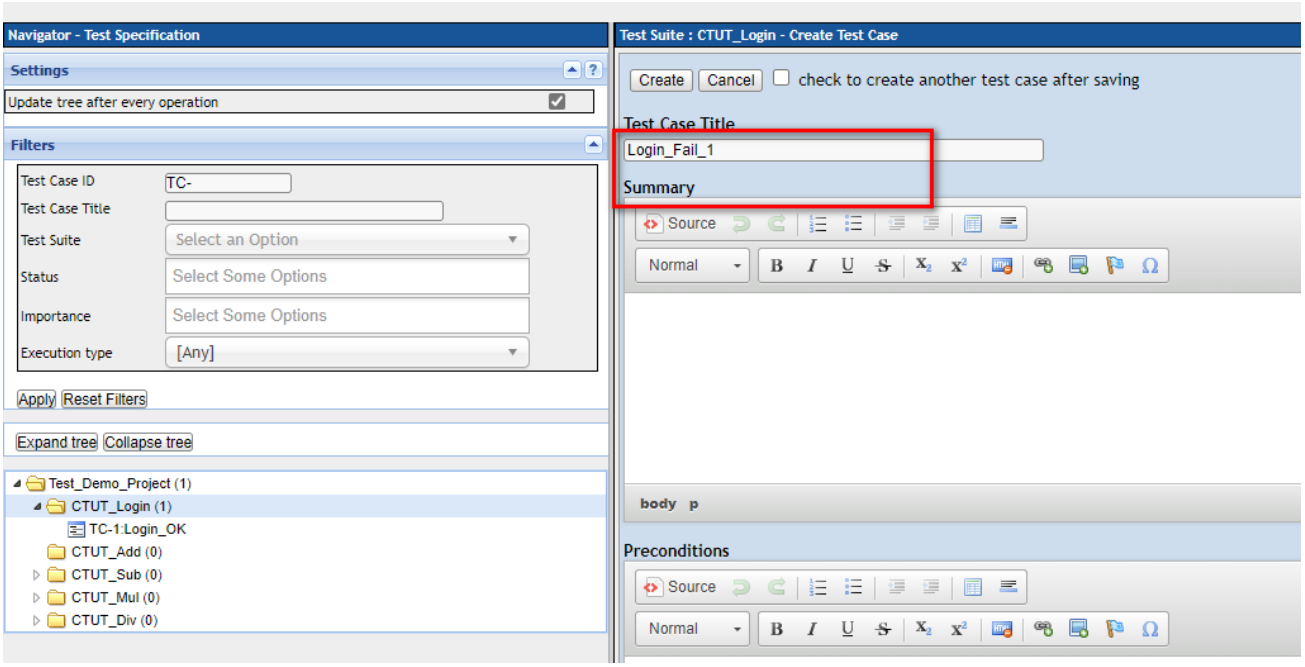
3.5. Tạo Test Case

Sau khi tạo (Test Suite), chúng ta phải tạo Test Case cho bộ Test Suite đó. Số lượng Test Case trong Test Suite phụ thuộc vào Requitements có quy định số Test Case, do đó chúng ta phải tạo Test Case cho từng yêu cầu. Cách thực hiện như sau:

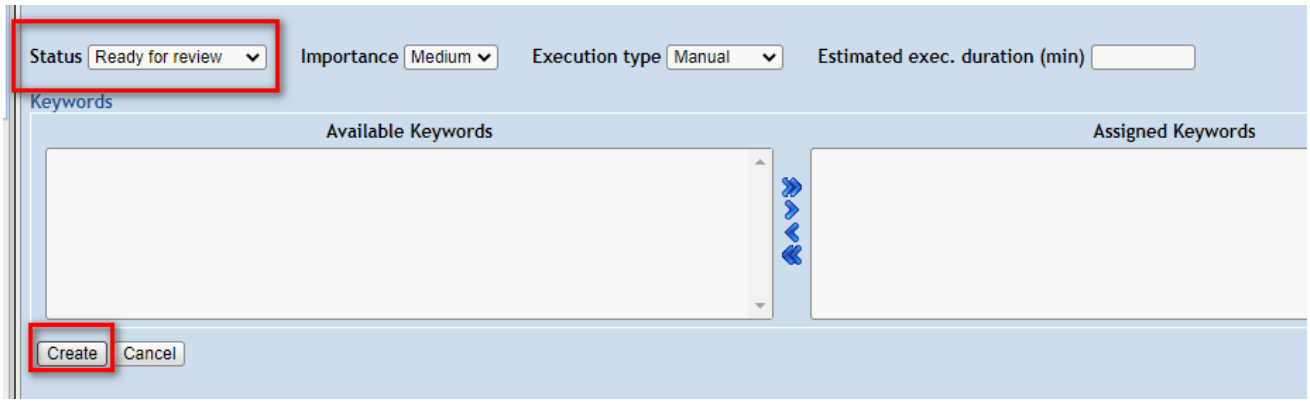
- Bước 1: Chọn yêu cầu -> **Create Test Cases** như hình 31.



Hình 31: Tạo mới Test Case

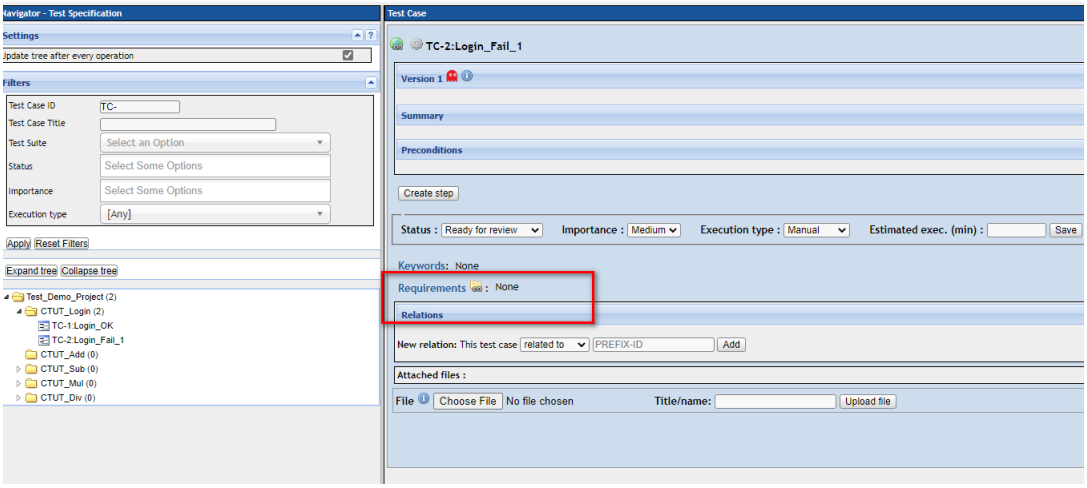


Hình 32: Thêm thông tin Test Case

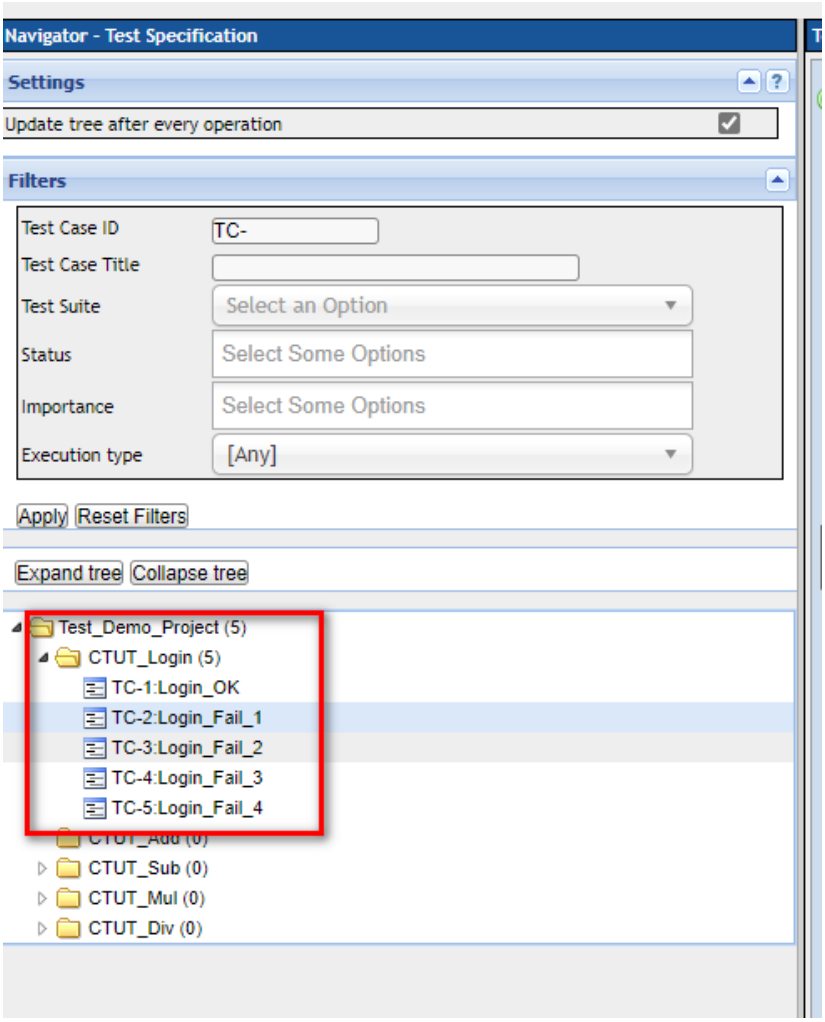


Hình 33: Hoàn tất tạo Test Case

- Bước 2: Gán Test Case cho Requirement như sau:

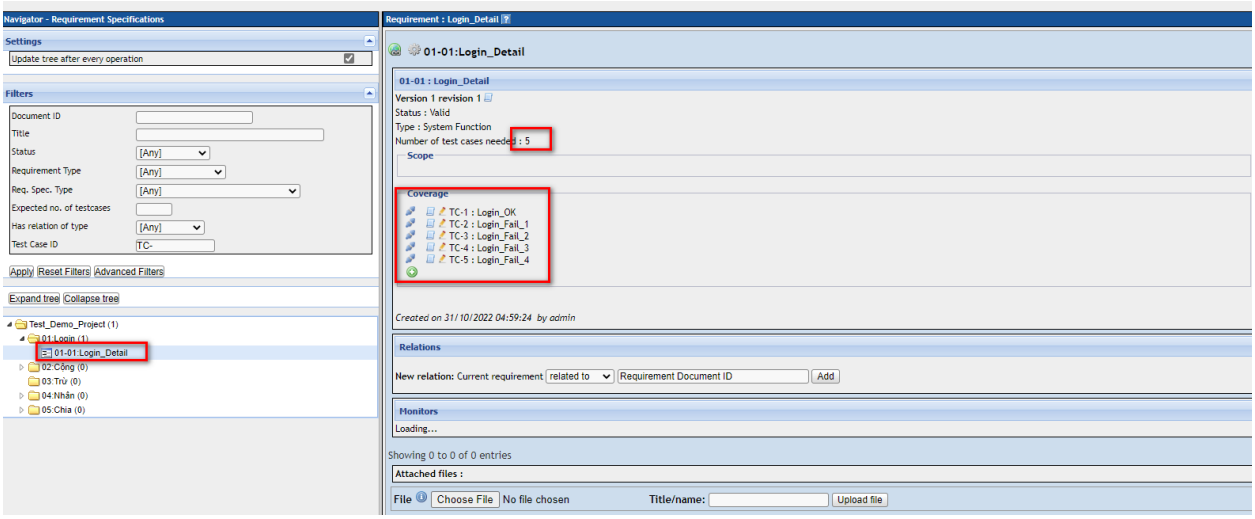


Hình 34: Gán Test Case cho Requirement



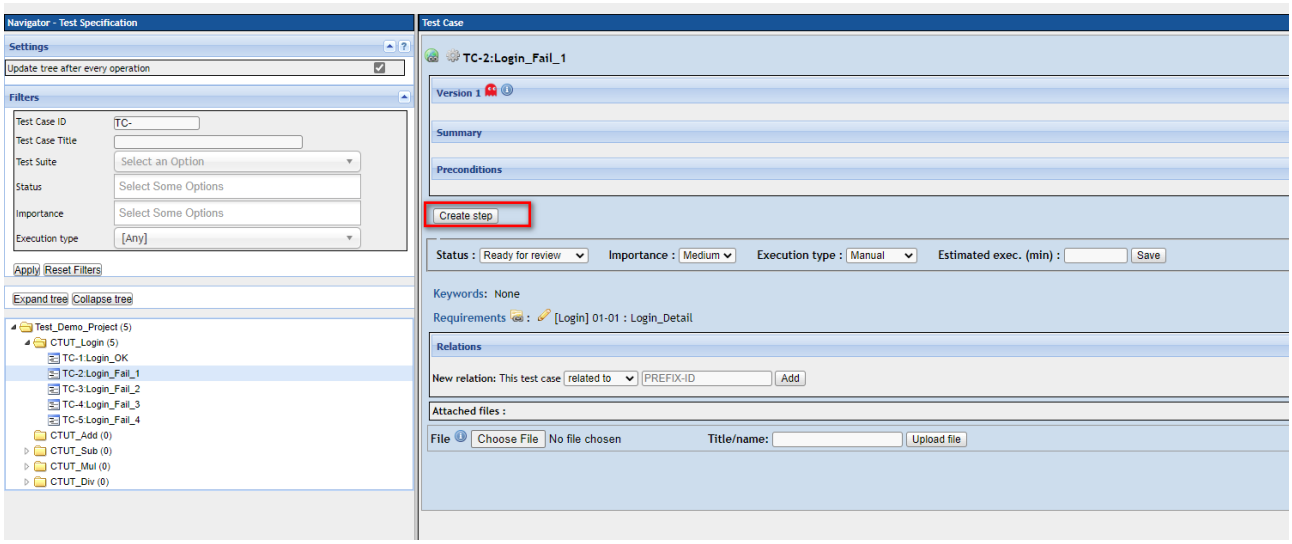
Hình 35: Danh sách Test Case

- Bước 3: Kiểm tra link Test Case với Requirement.

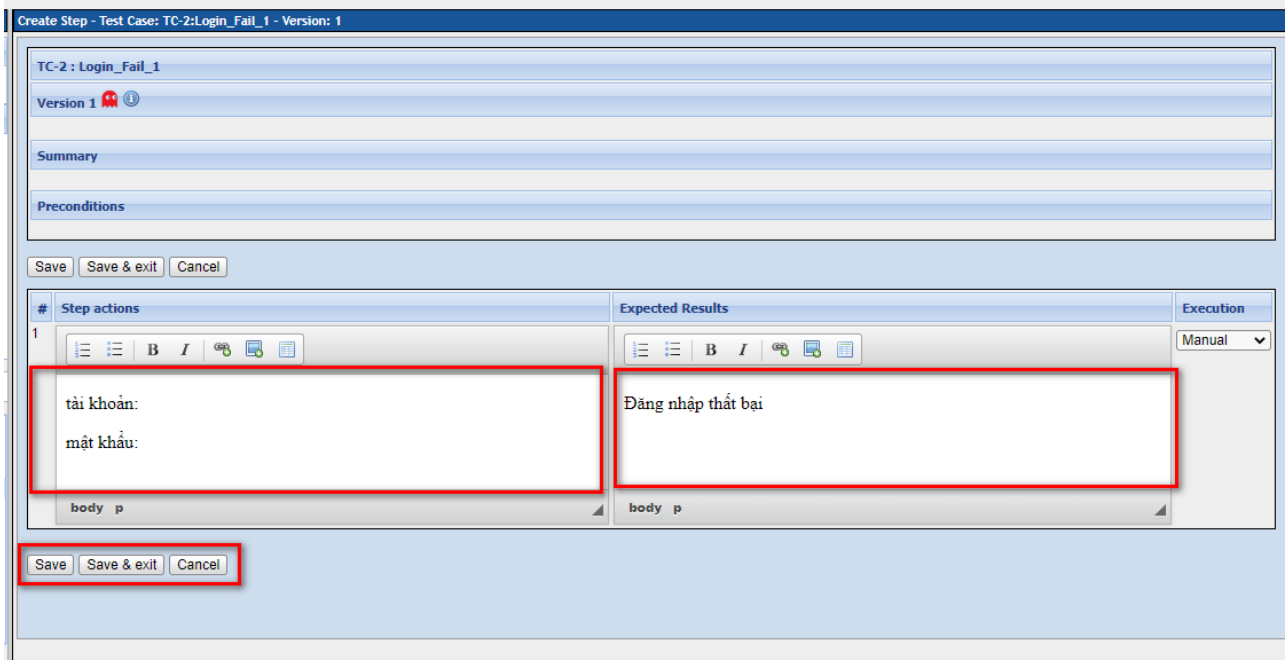


Hình 36: Danh sách Test Case của Requirement

- Bước 4: Tạo step và data cụ thể cho Test Case. Vào mục Test specification, chọn Test Case và ấn nút Create step.



Hình 37: Tạo step cho Test Case



Hình 38: Lưu step

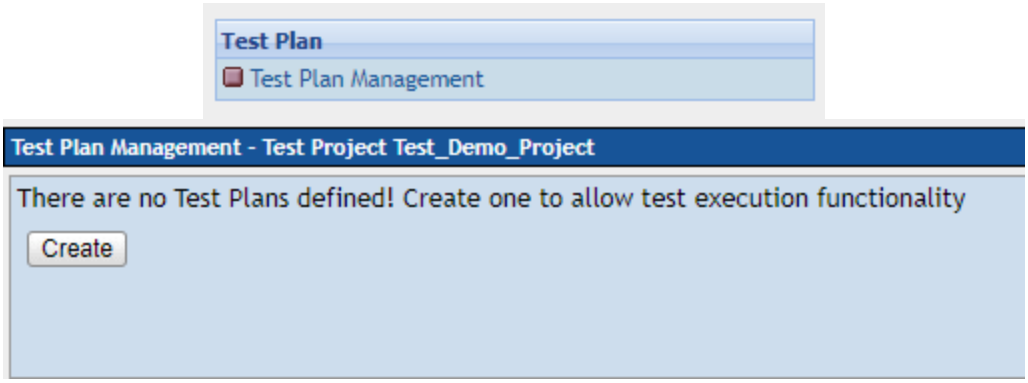
3.6. Tạo Test Plan

Test Plan chứa đầy đủ thông tin như phạm vi kiểm thử, các mốc kiểm thử, Test Suites và Test Cases. Thường khi đã tạo xong Project Test thì bước tiếp theo là tạo Test Plan. Phương pháp tạo Test Plan trong TestLink được nhóm tác giả trình bày chi tiết trong các nội dung trình bày bên dưới.

3.6.1. Tạo mới Test Plan cho Project

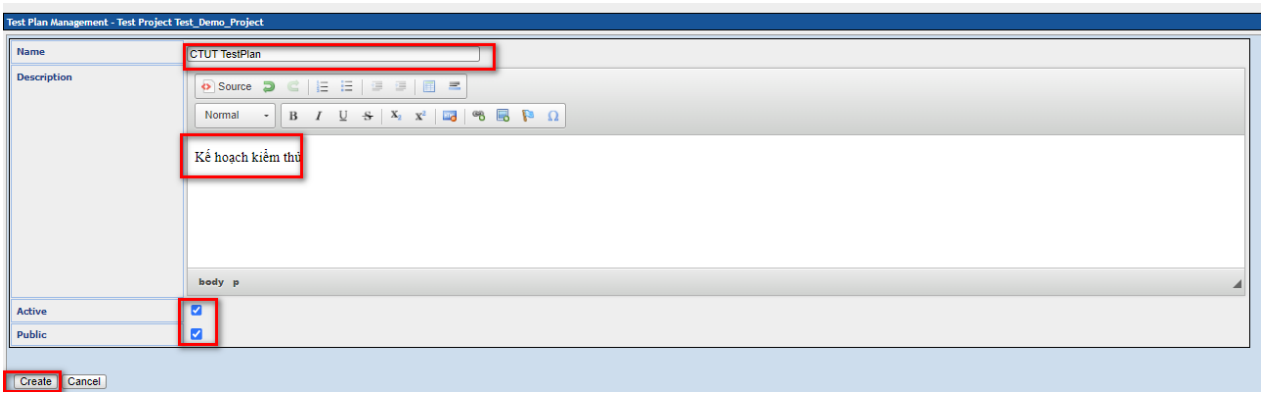
- Bước 1: Ấn vào Project trên Menu TestLink hoặc logo TestLink.

- Bước 2: Chọn Test Plan Management, ấn **Create**.

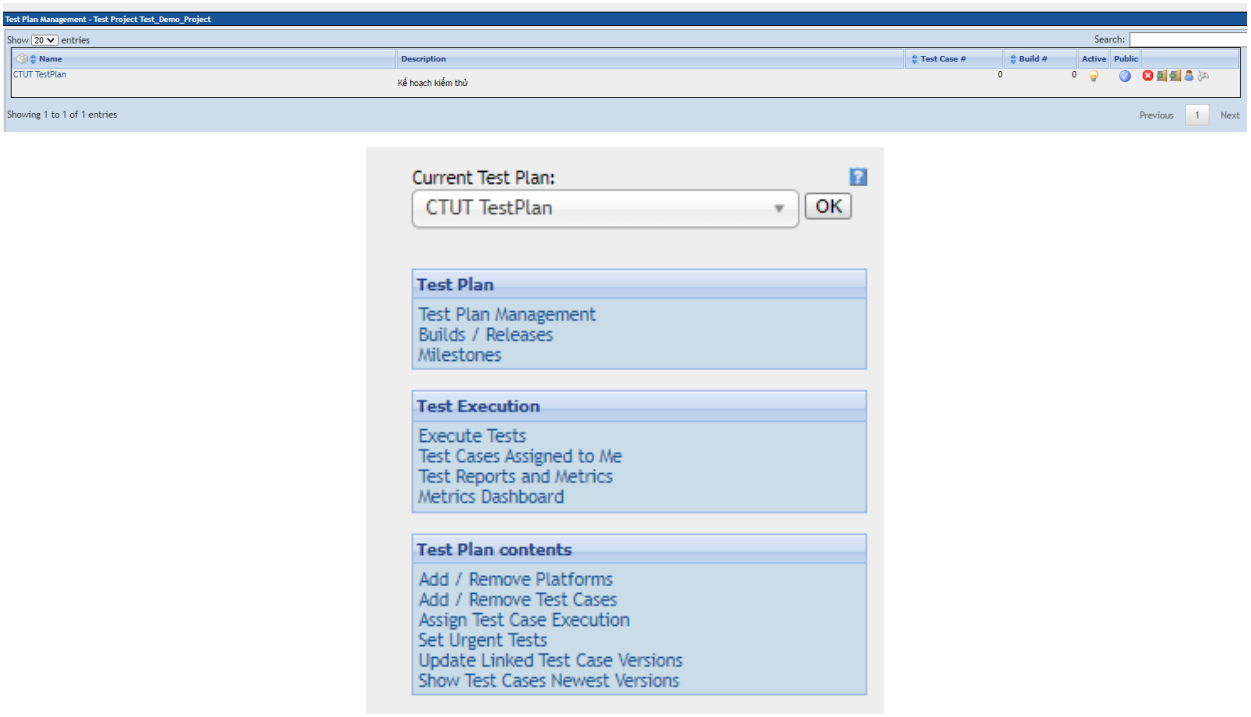


Hình 39: Tạo mới Test Plan

- Bước 3: Thêm thông tin như hình 40. Ấn nút Create. Kết quả như Hình 41.



Hình 40: Thông tin tạo Test Plan

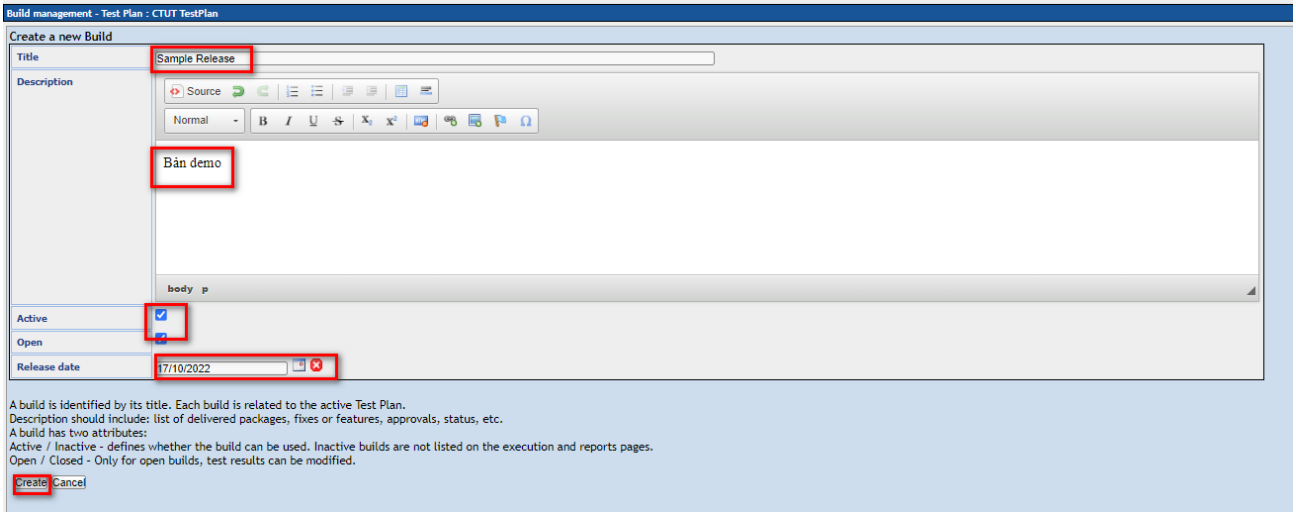


Hình 41: Thông tin Test Plan

3.6.2. Builds/Releases

Cung cấp chức năng cài đặt ngày hoàn thành kiểm thử và thực hiện Releases sản phẩm cho khách hàng. Chung ta chọn **Builds/Releases -> Create**. Sau đó, nhập vào tất cả các trường cần thiết cho bản sản phẩm được phát hành và ấn vào nút **Create**, các thông tin yêu cầu nhập vào được thực hiện tuần tự như sau:

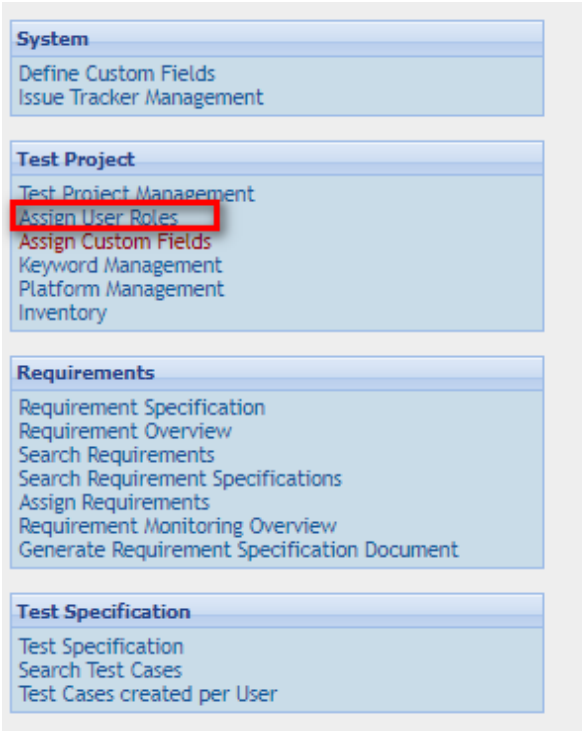
- ✓ Nhập tên Title
- ✓ Nhập Description cho bản phần mềm được release
- ✓ Check vào ô Active
- ✓ Check vào ô Open
- ✓ Chọn ngày release sản phẩm
- ✓ Click vào button "Create"

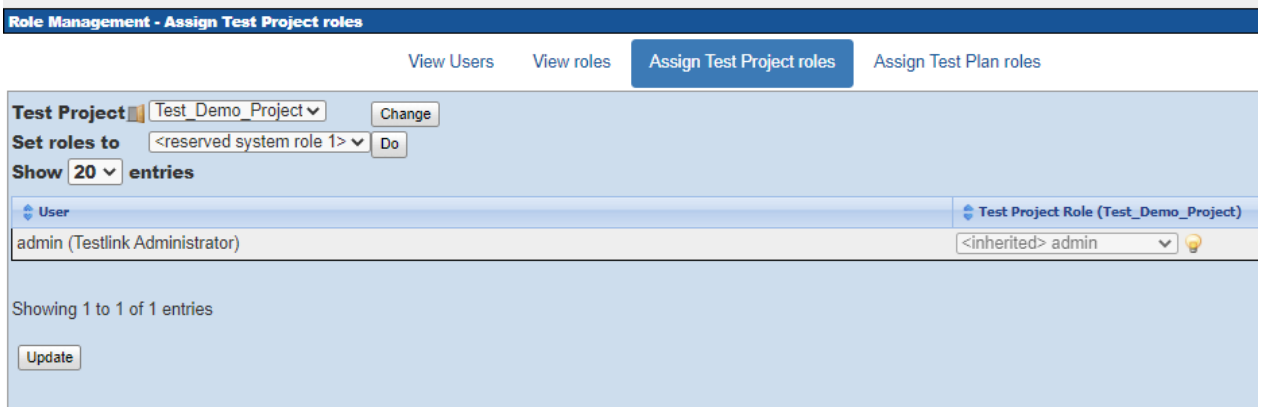


Hình 42: Thông tin tạo Releases

3.6.3. Assign User Roles

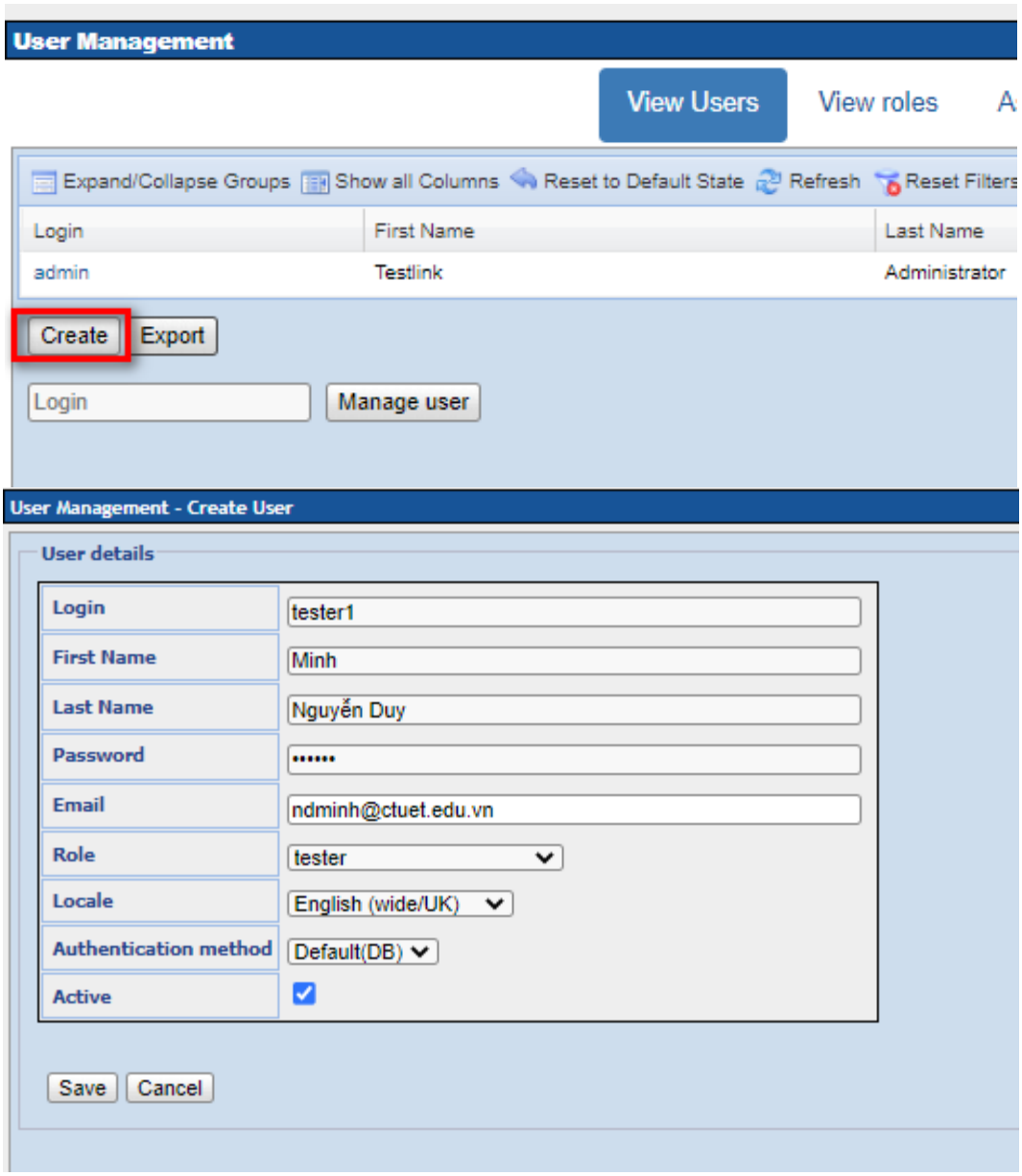
Thực hiện thêm các đối tượng vào Test Plan như: Tester, Test designer, Admin… Muốn thực hiện tốt chức năng này thì chúng ta phải tạo người dùng trước bằng cách sử dụng chức năng **User Management** như Hình 43.





Hình 43: Giao diện quản lý User

Ấn nút **Create** trong chức năng **View Users**, điền thông tin như Hình 44, ấn nút **Save** để lưu lại thông tin.

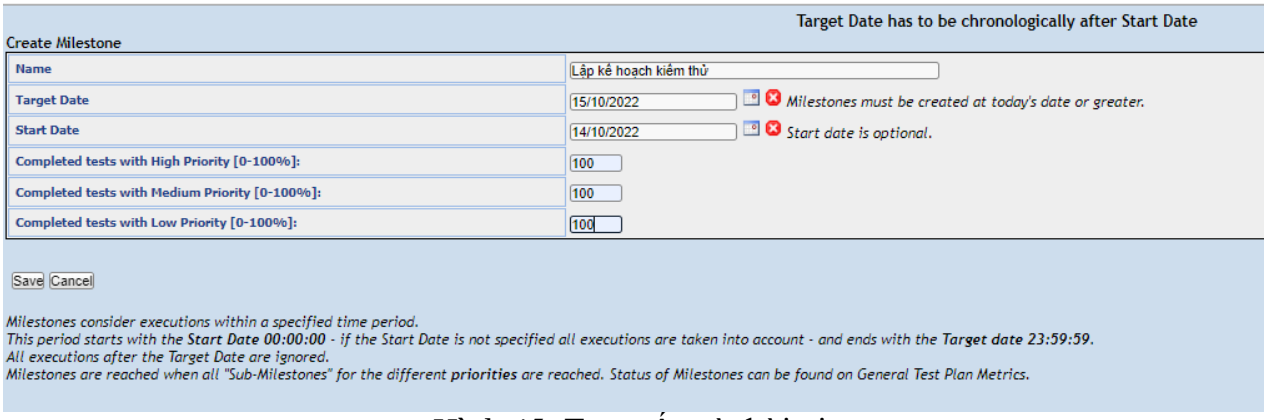


Hình 44: Thông tin User mới

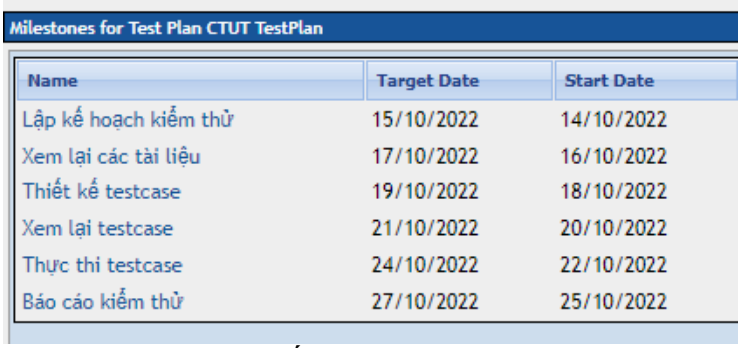
Chúng ta có thể cài đặt lại các quyền cho các đối tượng trong chức năng **View Roles**. Gán quyền cho các đối tượng trong Project, Test Plan thông qua chức năng Assign Test Project Roles/Assign Test Plan Roles.

3.6.4. Milestones Overview

Chúng ta thiết kế mốc thời gian và công viên trong quản lý kiểm thử của Test Plan thông qua chức năng này. Chọn Create để tạo Milestones.



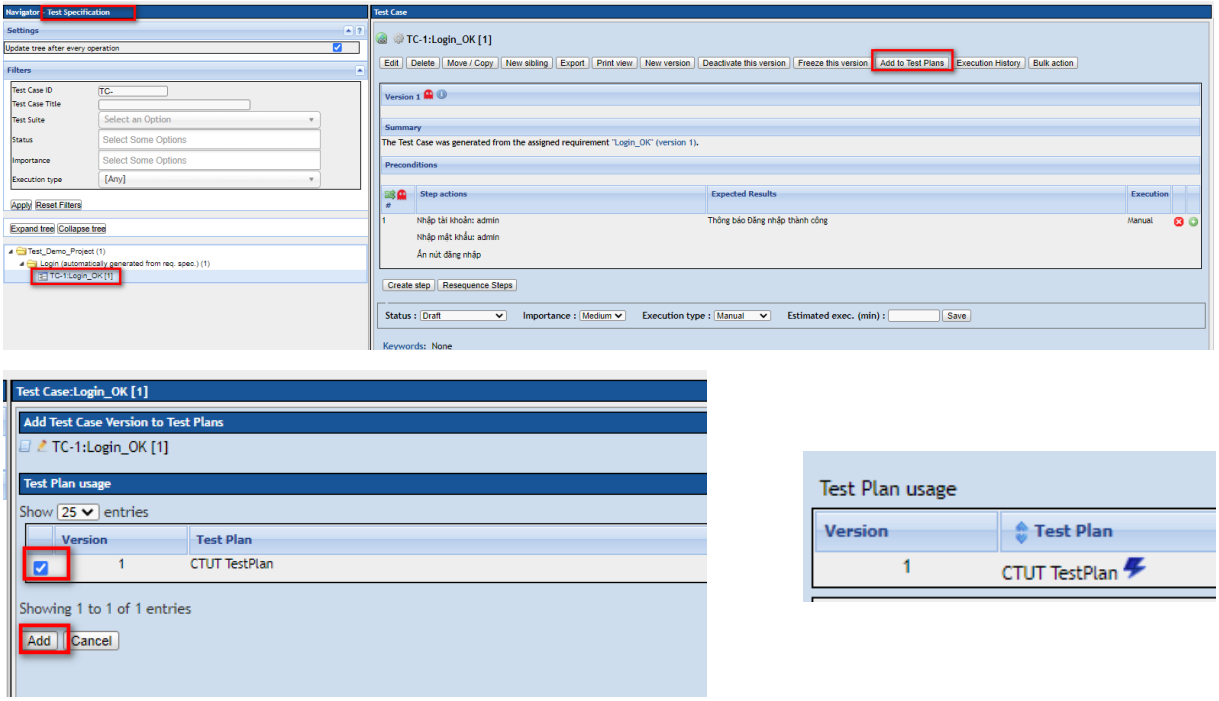
Hình 45: Tạo mốc và thời gian



Hình 46: Các mốc thời gian trong Test Plan

3.6.5. Gán Test Case vào Test Plan

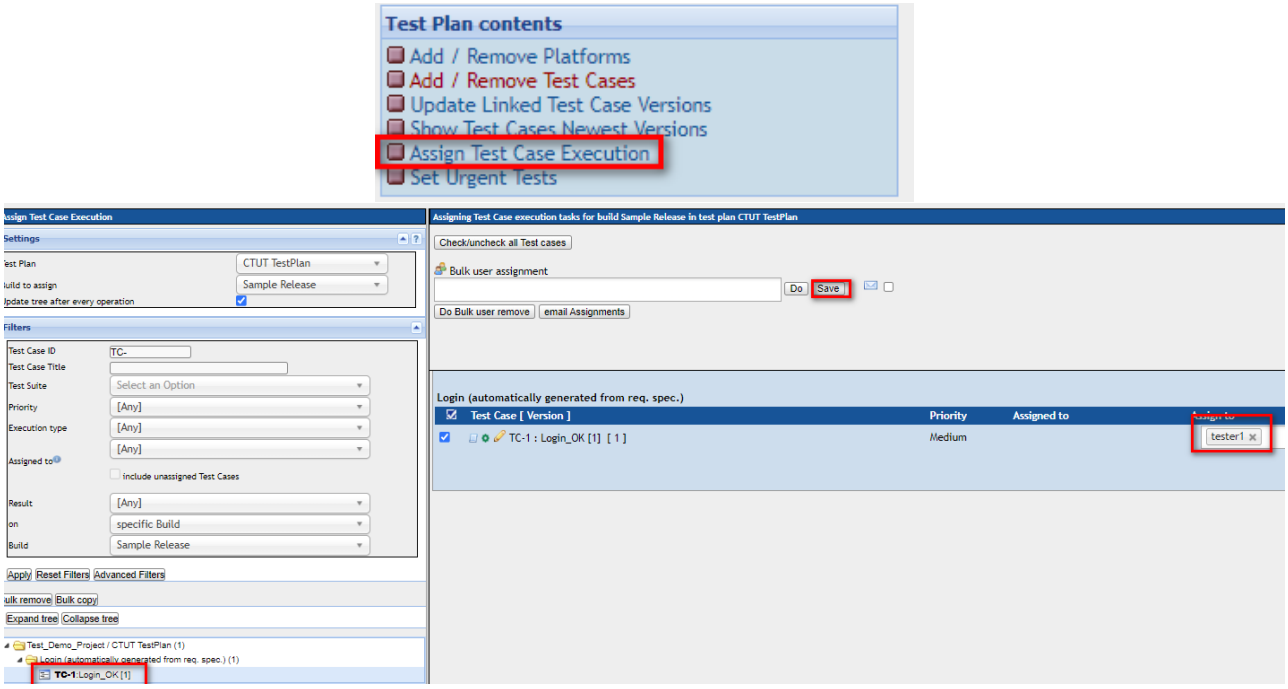
Sau khi tạo Test Plan, chúng ta thực hiện các Test Case cho Test Plan đã tạo. Sử dụng chức năng Test Specification -> Chọn Test Case -> Add to Test Plans.



Hình 47: Thêm Test Case vào Test Plan

3.6.6. Gán Test Case cho Tester

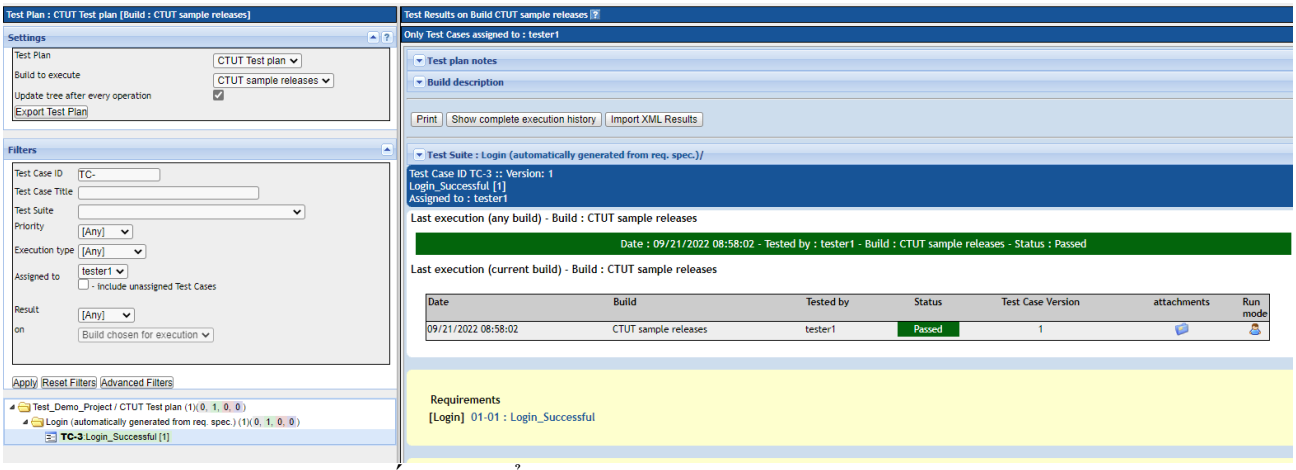
Để giao nhiệm vụ kiểm thử các Test Case cho các Tester trong Test Plan, sử dụng chức năng Assign Test Case Execution như hình 48.



Hình 48: Gán Test Case cho Tester

3.7. Thực thi Test Case

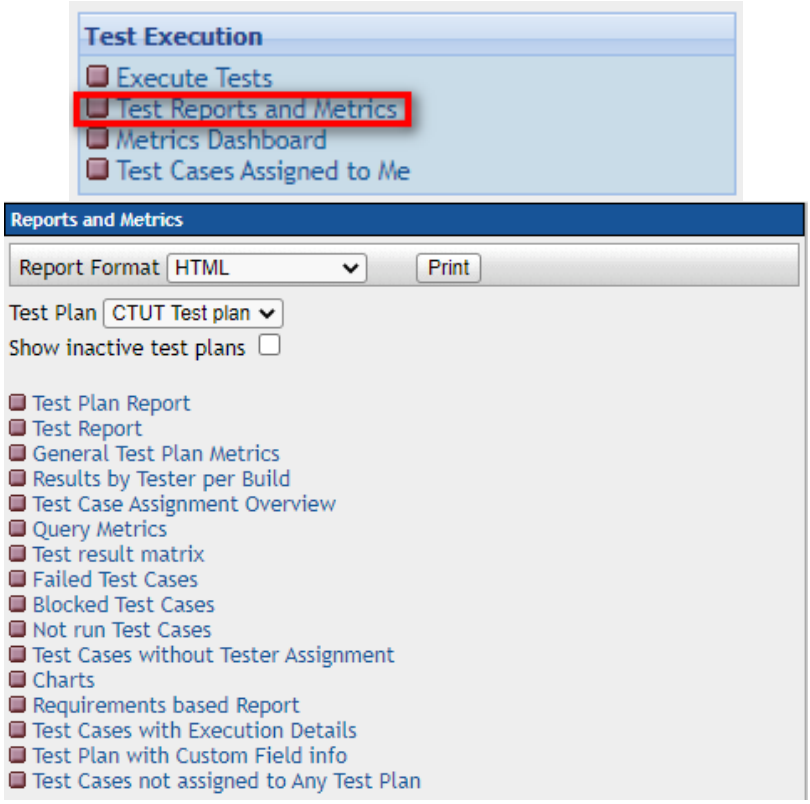
Đăng nhập tài khoản đã tạo và sử dụng chức năng **Test Execution** để thực thi kiểm thử, ghi kết quả để tạo Test Report. Nếu Fail có thể thêm Notes/Description để Developer sửa lỗi hiệu quả.



Hình 49: Kết quả kiểm thử Test Case được gán cho Tester1

3.8. Tạo Test Report

Để tạo Test Report, chúng ta sử dụng chức năng **Test Report and Metrics** của nhóm Test Execution thuộc Test Plan.



Hình 50: Các chức năng trong Report and Metrics

- Test Plan Report: Báo cáo thông tin chung về Test Plan.

1.1. Login (automatically generated from req. spec.)

Login_Successful [1]

Scope

Kế hoạch kiểm thử phần mềm

1.1. Test Suite : Login (automatically generated from req. spec.)

Test Cases in the Test Suite are generated from Requirements. A refinement of test scenario is highly recommended.

Test Case TC-3: Login_Successful [1]		
Author:	admin	
Summary:		
The Test Case was generated from the assigned requirement.		
Đăng nhập đúng, hiện thị giao diện chính		
Preconditions:		
- Kích hoạt phần mềm. - Được cung cấp tài khoản đăng nhập.		
#:	Step actions:	Expected Results:
1	1. Nhập tài khoản 2. Nhập mật khẩu 3. Ấn nút đăng nhập	Đăng nhập thành công. Hiển thị giao diện chính
Requirements	01-01: Login_Successful	
Keywords:	None	

Hình 51: Thông tin chung Test Plan

➢ Test Report: Báo cáo chi tiết kết quả các Test Case theo yêu cầu.

1.1. Login (automatically generated from req. spec.)

Login_Successful [1]

1.1. Test Suite : Login (automatically generated from req. spec.)

Test Cases in the Test Suite are generated from Requirements. A refinement of test scenario is highly recommended.

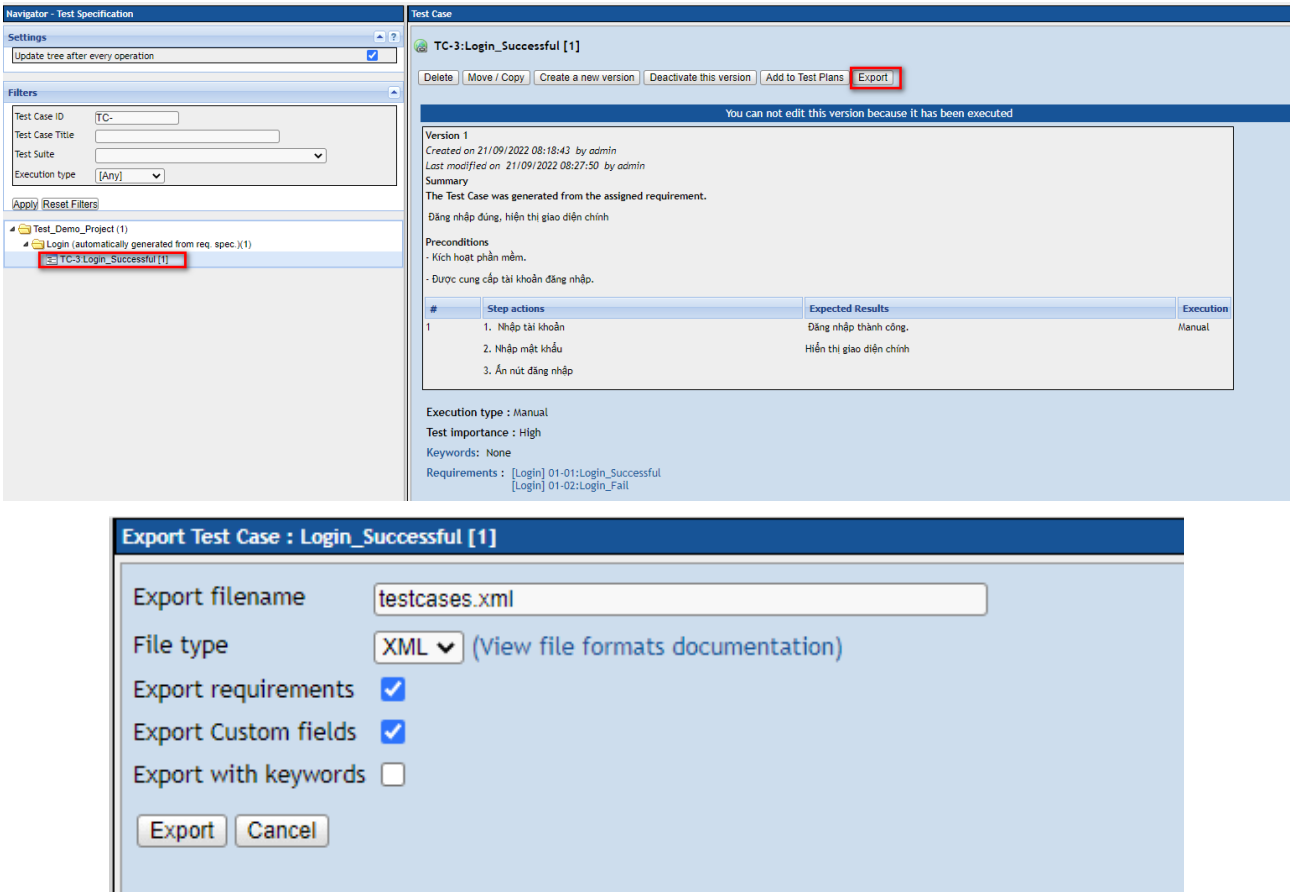
Test Case TC-3: Login_Successful [1]		
Author:	admin	
Summary:		
The Test Case was generated from the assigned requirement.		
Đăng nhập đúng, hiện thị giao diện chính		
Preconditions:		
- Kích hoạt phần mềm. - Được cung cấp tài khoản đăng nhập.		
#:	Step actions:	Expected Results:
1	1. Nhập tài khoản 2. Nhập mật khẩu 3. Ấn nút đăng nhập	Đăng nhập thành công. Hiển thị giao diện chính
Last Result:	**Passed**	
Build	CTUT sample releases	
Tester	ester1	
Requirements	01-01: Login_Successful	
Keywords:	None	

Hình 52: Chi tiết thực thi Test Case

3.9. Export và Import Test Case

3.9.1. Export to XML

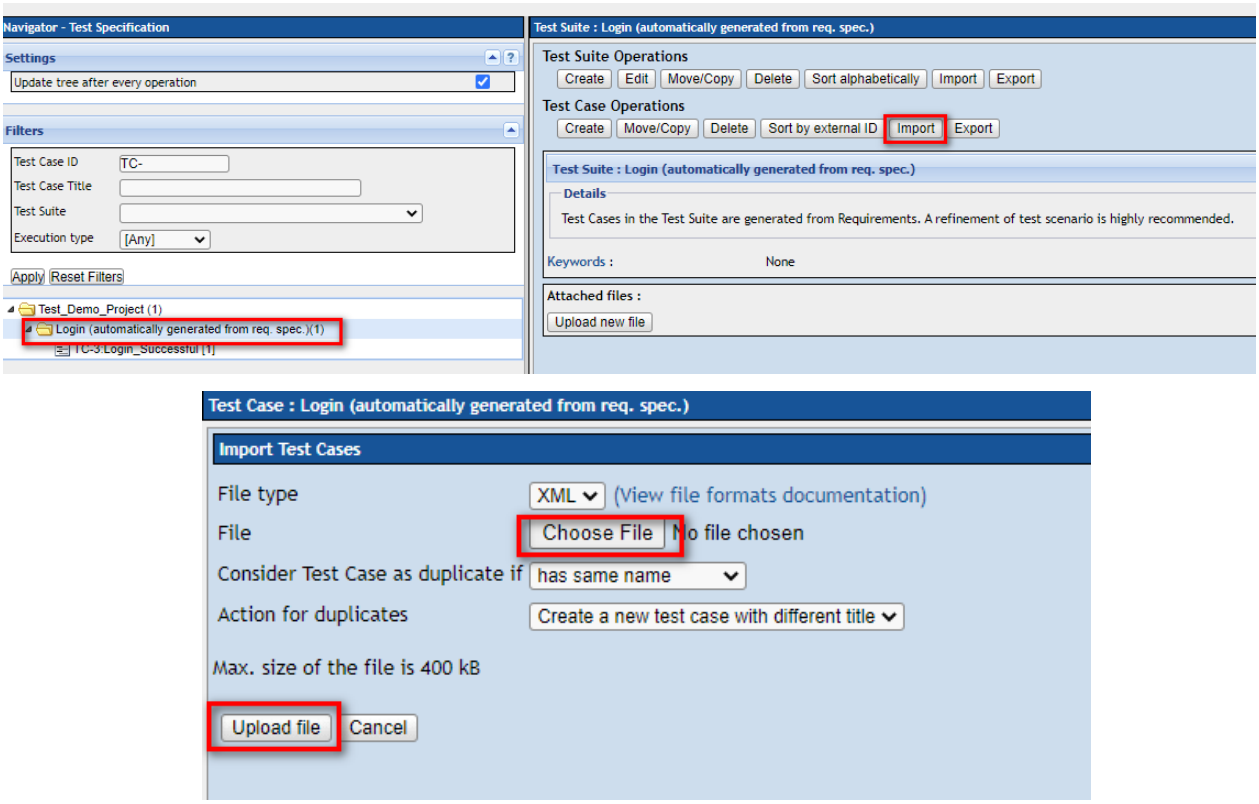
Để xuất thông tin Test Case từ TestLink, chúng ta sử dụng chức năng Test Specification. Sau đó chọn Test Case và ấn Export như hình 53.



Hình 53: Export Test Case

3.9.2. Import

Để import Test Case vào TestLink, chúng ta chọn Test Specification và chọn nút Import như hình 54. Chọn đường dẫn để upload file XML của Test Case.



Hình 54: Import Test Case